


RESEARCH

Open Access



Shorter ZK-SNARKs from square span programs over ideal lattices

Xi Lin^{1,2}, Heyang Cao^{1,2*} , Feng-Hao Liu³, Zhedong Wang⁴ and Mingsheng Wang^{1,2}

Abstract

Zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs) are cryptographic protocols that offer efficient and privacy-preserving means of verifying NP language relations and have drawn considerable attention for their appealing applications, e.g., verifiable computation and anonymous payment protocol. Compared with the pre-quantum case, the practicability of this primitive in the post-quantum setting is still unsatisfactory, especially for the space complexity. To tackle this issue, this work seeks to enhance the efficiency and compactness of lattice-based zk-SNARKs, including proof length and common reference string (CRS) length. In this paper, we develop the framework of square span program-based SNARKs and design new zk-SNARKs over cyclotomic rings. Compared with previous works, our construction is without parallel repetition and achieves shorter proof and CRS lengths than previous lattice-based zk-SNARK schemes. Particularly, the proof length of our scheme is around 23.3% smaller than the recent shortest lattice-based zk-SNARKs by Ishai et al. (in: Proceedings of the 2021 ACM SIGSAC conference on computer and communications security, pp 212–234, 2021), and the CRS length is 3.6× smaller. Our constructions follow the framework of Gennaro et al. (in: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 556–573, 2018), and adapt it to the ring setting by slightly modifying the knowledge assumptions. We develop concretely small constructions by using module-switching and key-switching procedures in a novel way.

Keywords Zk-SNARKs, Post-quantum, Succinct argument

Introduction

Zero-knowledge (ZK) proofs are cryptographic protocols that enable a prover to persuasively demonstrate the validity of a specific statement to a verifier while keeping the witness secret. The concept was initially introduced by Goldwasser et al. (1989), and there have been active researches in both theory and practice since then.

In numerous scenarios, it is essential for the prover to genuinely possess knowledge of a valid witness, thereby establishing an argument of knowledge. To enhance efficiency, specific characteristics like non-interactive and succinctness are highly desirable. These proofs entail a single round of message exchange from the prover's side, enabling the verifier to validate the correctness in a considerably shorter time compared to the prover's computational effort. These attributes give rise to a class of cryptographic constructions, commonly known as succinct non-interactive arguments of knowledge (ZK)-SNARKs. It finds wide-ranging applications, including verifiable computations (Ben-Sasson et al. 2013, 2014; Parno et al. 2016) and anonymous payment protocols (Sasson et al. 2014). Despite these compelling features, some negative results are associated with these constructions. Gentry and Wichs (2011) demonstrated that no

*Correspondence:

Heyang Cao
caoheyang@iie.ac.cn

¹ Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

² School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

³ School of Electrical Engineering & Computer Science, Washington State University, Pullman, WA, USA

⁴ School of Cyber Science and Engineering, Shanghai Jiaotong University, Shanghai, China

secure succinct non-interactive arguments (SNARGs) existed in the standard model. Consequently, all existing SNARGs are constructed in the Random Oracle Model or rely on non-falsifiable assumptions (Naor 2003). Additionally, the most efficient SNARKs are designed verifiers, wherein only those who possess the verification keys are authorized to validate the proofs, in contrast to the public verifiers that permit anyone to verify a proof.

The concept of SNARK has been extensively investigated in the literature (Bitansky et al. 2011, 2012, 2017; Goldwasser et al. 2011), and subsequent works mainly focus on enhancing the efficiency for practical use. The early schemes (Gennaro et al. 2013; Danezis et al. 2014) in this area were almost based on group or bilinear pairing. Nowadays, driven by the advances in quantum computation and quantum computers, post-quantum security progressively attracts more attention. Many lattice-based SNARKs have emerged in recent years.

However, the lattice-based constructions have a significant inefficiency compared to the group or pairing-based ones. Intuitively, the optimal scheme belongs to preprocessing SNARK and was proposed by Groth (2016), whose proof length is 128B. The state-of-the-art post-quantum SNARK was proposed by Ishai et al. (2021), whose proof size is 16.4KB, which is 131.2x larger. Furthermore, as almost all efficient SNARKs necessitate a trusted setup, the length of the common reference string (CRS) also merits attention. Therefore, how to promote the efficiency of lattice-based SNARKs is an important and meaningful research problem.

These motivate our main question:

Can we improve the efficiency of lattice-based SNARKs, especially in the proof length and CRS length?

Related works

The constructions of SNARKs exhibit diverse design routes. Two paradigmatic routes are presented: one research line adopts a combination of polynomial interactive oracle proof (polynomial IOP) and the polynomial commitment; another research line is built on the circuit directly. The former approach presents a notable advantage in terms of applicability, such as transparent setup and public verifier, albeit at the expense of efficiency. On the contrary, the latter approach imposes certain limitations, requiring a trusted setup and designed verifier, but achieves higher efficiency.

The same applies to lattice-based SNARKs. Recent advancements in lattice-based SNARKs can be divided into two categories. For the first research line, the researcher tried to obtain SNARKs with attractive properties or functionalities. The most critical components

are various commitments, i.e., vector commitments (Peikert et al. 2021; Albrecht et al. 2022), and functional commitments (Wee and Wu 2023; Fisch et al. 2023). Albrecht et al. (2022) proposed the first lattice-based SNARK construction from vector commitment, in which the verifier is public and has logarithmic complexity, and the construction is recursively composable. Cini et al. (2023) proposed the first lattice-based recursive folding protocol with a polylogarithmic-time verifier for linear relations and the first lattice-based succinct argument with a linear-time prover for NP problem in the preprocessing model.

Before we review the lattice-based constructions following the second approach, we first retrospect the group-based ones. This route originated from Groth (2010), which constructed a non-interactive argument of zero-knowledge (NIZK) based on the circuit satisfiability problem. Then, the researchers found it is possible to convert the circuit satisfiability problem into more algebraic formulations to construct efficient SNARKs. Many works introduced different characterizations of the NP complexity class: quadratic span programs (QSPs) (Gennaro et al. 2013), square span programs (SSPs) (Danezis et al. 2014), and rank-1 constraint systems (R1CS) (Ben-Sasson et al. 2013). Then many efficient constructions of SNARKs based on specific structures came. Detailedly, Gennaro et al. (2013) proposed constructions based on QSPs, whose proof consists of 7 group elements and the CRS size is linear in the circuit size. In the next year, Danezis et al. (2014) introduced SSPs and built SNARKs based on SSPs (a simpler form than QSPs), whose proof consists of 4 group elements. Meanwhile, a concurrent research line (Bitansky et al. 2013; Boneh et al. 2017) studied a more abstract cryptographic primitive: linear probabilistically checkable proof (LPCP). They established constructions of LPCP for NP problems and then built SNARK (SNARK) based on LPCP. The nature of the above designs can be unified in that preprocessing implies holography as claimed in Chiesa and Yorgev (2020), but the revealing information of probabilistically checkable proof differs.

In terms of efficiency, the optimal scheme belongs to preprocessing and designated-verifier SNARKs and was proposed by Groth (2016), whose proof only consists of 3 group elements. Its proof length is 128B for the circuit of size 2^{20} , which significantly outperforms other schemes. This is also the most widely used SNARK scheme in practice, i.e., ZCash (Sasson et al. 2014), Filecoin (Labs Labs 2018), and Coda (Bonneau et al. 2020).

In the domain of lattice-based SNARKs, Boneh et al. (2017) introduced the first quasi-optimal SNARKs based on lattice, employing linear multi-prover interactive

proofs. Closely followed by this work, Gennaro et al. (2018) put forward the first lattice-based SNARK scheme, which was built on SSPs. Nitulescu (2019) introduced the first lattice-based zk-SNARG for arithmetic circuits leveraging square arithmetic programs (SAPs), whose proof consists of 2 LWE ciphertexts. Naganuma et al. (2020) proposed faster zk-SNARK constructions for arithmetic circuits using quadratic arithmetic programs (QAPs), whose proof consists of 3 LWE ciphertexts. Then, Ishai et al. (2021) followed the framework of Bitansky et al. (2013) and Boneh et al. (2017) and proposed a new LPCP-based SNARK, which is the state-of-the-art parameters for lattice-based SNARKs. The most recent lattice-based SNARKs from Chung et al. (2023), proposed a new noise flooding technique and achieved smaller proof length in the amortized sense.

Our results

This research endeavors to tackle the aforementioned issue by devising novel, efficient SSP-based zk-SNARKs. Notably, we have succeeded in reducing proof and CRS lengths by circumventing parallel repetition, while retaining a high level of soundness. To provide a more comprehensive understanding of our work, we present a comparative analysis with prior research in Table 1. (It is essential to highlight that the estimation methodology employed in Ishai et al. (2021) is suboptimal, necessitating the adjustment of their parameters using the same “ADPS16” method to enable a more precise and reliable comparison. The CRS length is empty since they did not provide it.)

Table 1 Comparison of lattice-based SNARKs

Scheme	Circuit size	Proof length (KB)	CRS length
Gennaro et al. (2018)	2^{15}	640	8.63 MB
Nitulescu (2019)	2^{15}	270	–
Naganuma et al. (2020)	2^{15}	405	–
Chung et al. (2023)	2^{15}	319.5	–
Ishai et al. (2021)			
Shorter proof	2^{16}	17.54	191MB
	2^{20}	18.7	5.22GB
Shorter CRS	2^{16}	22.84	104MB
	2^{20}	23.83	1.9GB
Ours			
Basic	2^{16}	121.88	86.25MB
	2^{20}	130.53	1.41GB
Optimized	2^{16}	14.06	133.99MB
	2^{20}	14.34	1.48GB

Technical overview

Next, we present a summary of our technical contributions below.

Get Rid of Parallel Repetition by Ring Structure. Parallel repetition is a standard technique to amplify (knowledge) soundness error. In the field (\mathbb{Z}_p or even \mathbb{Z}_{p^2}), if we do not use parallel repetition and guess a random element over the field with probability lower than 2^{-128} , it requires the modulus p satisfies that $p > 2^{128}$ (or $p^2 > 2^{128}$), which is too large. Therefore, previous works chose smaller p (such as 32-bits or 19-bits) and use parallel repetition for a desired security level.

To deal with this issue, we adopt a strategy of transforming the field structure into a ring structure. To illustrate, if we consider a ring with the modulus p and dimension n , the desired target can be accomplished by ensuring that $p^n > 2^{128}$. Albeit combining with other limitations in our construction, the final requirement turns out to be $2d/p^{\frac{n}{2}} < 2^{-128}$. However, solely employing the ring structure may not suffice in reducing the parameter size and may potentially incur additional issues. As such, supplementary techniques must be employed to tackle these issues, which will be expounded upon below.

Reductions from Boolean Circuits over Ring. Both SSP-based schemes and LPCP-based schemes use polynomial interpolation to express circuits into SSP/LPCP instances. Prior works (to our knowledge) consider polynomial interpolation over fields, and extending it to the rings inheres challenges, particularly with regards to invertibility in R . Towards this, we leverage a useful result (Katsumata and Yamada 2016), which stated that the ring elements with a “small” norm are invertible. More concretely, in the polynomial interpolation, the denominators of the interpolation coefficients take the form of $x_i - x_j$ for distinct i, j . In order to ensure that $x_i - x_j$ has an inverse over R_p , we restrict the domain of x_i and x_j to $R_{\{0,1\}}$, where the coefficients of polynomials are either 0 or 1. As a result, we can instantiate polynomial interpolation over the ring of our choice.

Optimizations via Ciphertext Operations. As noted above, the SSP-based scheme presented in Gennaro et al. (2018) has a large proof length, primarily due to its inclusion of five ciphertexts in the proof. In contrast, the LPCP-based scheme proposed by Ishai et al. (2021) utilizes different encrypted queries as the CRS, which are multiplied by the same coefficients during proof generation. This allows for the utilization of the packing method described in Peikert et al. (2008) to reduce the proof length by sharing randomness. Unfortunately, the SSP-based scheme involves different coefficients (e.g., \mathbf{h}, \mathbf{v}), which precludes the direct application of the aforementioned method. However, in the ring setting, we can

leverage the ring structure to pack the 5 ciphertexts into a single ciphertext. This approach reduces the number of ciphertexts for constructing the proof.

The utilization of a packing technique leads to a decrease in the number of ciphertexts, although it comes at the expense of augmenting the ring dimension. This implies that the size of the proof has not undergone any reduction. To address this, we employ the key-switching technique to attain a shorter proof. As a consequence, a slight modification of the knowledge assumption becomes necessary. Further deliberations are provided in section "Assumptions".

Preliminaries

Basic notations and probability results

Let λ, κ represent the computational, and statistical security parameters respectively. The negligible function $\text{negl}(\lambda)$ is strictly bounded by $1/\lambda^c$ for large λ , constant $c > 0$. On the contrary, the overwhelming probability represents the value to be $1 - \text{negl}(\lambda)$.

In our notation, a bold lowercase letter (e.g., \mathbf{x}) signifies a column vector, while a bold uppercase letter (e.g., \mathbf{A}) represents a matrix.

\mathbb{Z} represents the set of integers, and \mathbb{Z}_q indicates the ring of integers modulo q . R is a polynomial ring, and R_q indicates the ring elements in R modulo q . Then we adopt the unified notation $[a]_q$ to represent $a \bmod q$ encompassing both integer and ring elements, without distinction. In the case where the modulus q is not a power of 2, we employ $\log q$ to substitute $\lceil \log_2 q \rceil$ for simplicity.

We use $u \stackrel{\$}{\leftarrow} U$ to indicate that sample a random element u from the set U . For two distributions A, B , let $A \stackrel{s}{\approx} B, A \stackrel{c}{\approx} B$ represent statistically close, computationally indistinguishable respectively.

Gaussian Distribution. The n -dimension Gaussian function with parameter $\sigma > 0$ is defined as $\rho_\sigma(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|_2^2 / \sigma^2)$. Based on this, the discrete Gaussian distribution over \mathbb{Z}^n is defined as $D_{\mathbb{Z}^n, \sigma} = \rho_\sigma(\mathbf{x}) / \rho_\sigma(\mathbb{Z}^n)$, where $\rho_\sigma(\mathbb{Z}^n) = \sum_{\mathbf{x} \in \mathbb{Z}^n} \rho_\sigma(\mathbf{x})$.

Lemma 1 (Banaszczyk (1995), Lemma 2.4) For any $s, t > 0$ and a integer vector $\mathbf{a} \in \mathbb{Z}^n$, we have $\Pr[\langle \mathbf{a}, D_{\mathbb{Z}^n, s} \rangle \geq ts \|\mathbf{a}\|_2] \leq 2 \exp(-\pi t^2 / s^2)$.

Schwartz-Zippel Lemma. Schwartz-Zippel lemma is commonly employed in the analysis of soundness error.

Lemma 2 \mathbb{F} is a finite field and K is a subset of \mathbb{F} (e.g., $K \subset \mathbb{F}$) with size $|K|$. Assume that the non-zero polynomial $f(Y_1, \dots, Y_n)$ has total degree D . If t_1, \dots, t_n are chosen from K randomly, then we have

$$\Pr [f(t_1, \dots, t_n) = 0] \leq \frac{D}{|K|}.$$

Cyclotomic rings

In this paper, we work on the power of 2 polynomial rings. Let n be a power of 2, and the $2n$ -th cyclotomic polynomial is defined as $\Phi_{2n}(x) = x^n + 1$. Then we define $2n$ -th cyclotomic ring as $R \cong \mathbb{Z}[x]/(x^n + 1)$ and the $16n$ -th cyclotomic ring as $\mathcal{R} \cong \mathbb{Z}[x]/(x^{8n} + 1)$. In this paper, we view ring elements via coefficient embedding. Namely, for any $s \in R$ we view $s = s_0 + s_1x + \dots + s_{n-1}x^{n-1}$ for $s_i \in \mathbb{Z}$. The ring addition and multiplication are with respect to modulo $x^n + 1$. Under the coefficient embedding, the ℓ_∞ and ℓ_2 norms for s are defined as: $\|s\|_\infty = \max_i \|s_i\|, \|s\|_2 = \sqrt{\|s_0\|^2 + \dots + \|s_{n-1}\|^2}$. Similarly, it is extended to the vector. For $\mathbf{a} = (a_1, \dots, a_t) \in R^t$, we define $\|\mathbf{a}\|_\infty = \max_i \|a_i\|_\infty, \|\mathbf{a}\|_2 = \sqrt{\|a_1\|_2^2 + \dots + \|a_t\|_2^2}$.

To discuss our choice of moduli, we first recall a special result from Katsumata and Yamada (2016).

Lemma 3 (Katsumata and Yamada (2016), Lemma 3) The prime p satisfies $p \bmod 8 = 3$ and n is a power of 2. Then $x^n + 1$ splits as $x^n + 1 = g_1 g_2 \bmod p$ with two irreducible polynomials in $\mathbb{Z}_p[x]$ $g_1 = x^{n/2} + vx^{n/4} - 1$ and $g_2 = x^{n/2} - vx^{n/4} - 1$, where $v^2 = -2 \bmod p$. Then, all $a \in R_p$ with $\|a\|_2 < \sqrt{p}$ are invertible.

MLWE problems and encoding schemes based on MLWE

Module-Learning with Error (MLWE). Module Learning with Error (Module-LWE) is a fusion of Ring-LWE and plain-LWE, which was proposed and studied in Brakerski et al. (2014); Langlois and Stehlé (2015). For the power of 2 cyclotomic rings, the ring R , and R^\vee only differ by a scale of n . Thus, we opt to work solely on R . More formally, the decision MLWE distribution and problem from Langlois and Stehlé (2015) are defined as follows:

Definition 4 (Module-LWE Distribution) Let ψ over R_q be the error distribution. Given a secret vector $\mathbf{s} \in R_q^k$, an instance in the MLWE distribution $A_{\mathbf{s}, \psi}$ over $R_q^k \times R_q$ is (\mathbf{a}, b) , where \mathbf{a} is chosen from R_q^k uniformly at random, e is from ψ , and $b = \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q$.

Definition 5 (Module-LWE, Decision Problem) The average-case decision MLWE $_{R_q, k, \psi}$ problem is to distinguish instances from $A_{\mathbf{s}, \psi}$ or from uniform distributions over $R_q^k \times R_q$.

The decision $\text{MLWE}_{R_q,k,\psi}$ problem is infeasible if for all ppt adversarys B given any polynomial number of samples, the probability that B solves $\text{MLWE}_{R_q,k,\psi}$ is negligibly close to $1/2$.

The Encoding Scheme. The encoding scheme used in the SNARK schemes can be symmetric and asymmetric. For convenience, we instantiate it as a symmetric MLWE scheme. Furthermore, the simple linear combination is not sufficient for zero-knowledge of SNARK, thus we re-randomize the linear evaluation procedure as that in Ishai et al. (2021).

Construction 6 (MLWE Encoding Scheme) For any positive integers n, k, Q , an encoding scheme MLWE with dimension n , rank k and modulus Q consists three ppt algorithms (K, E, D) and a randomized linear evaluation algorithm Eval. These algorithms are defined below:

- $K(1^\lambda, k)$: Sample $A^* \leftarrow R_Q^{k \times k}$, $\mathbf{s}', \mathbf{e}^* \leftarrow \Phi_\sigma^k$. Define $F = (A^*, b^*) = (A^*, (A^*)^T \mathbf{s}' + \mathbf{p}\mathbf{e}^*)$, $\mathbf{s} = (-\mathbf{s}', 1)$. Output (\mathbf{s}, F) .
- $E_s(m)$: Sample $\mathbf{a} \leftarrow R_Q^k$, and $e \leftarrow \Phi_\sigma$. Compute and output $\mathbf{c} = (\mathbf{a}, (\mathbf{s}', \mathbf{a}) + \mathbf{p}e + m)$.
- $\text{Eval}(\{\mathbf{c}_i = (\mathbf{a}_i, b_i), \alpha_i\}_{i \in [d]}, F)$: Sample independent $\mathbf{r}, \mathbf{e}' \leftarrow \Phi_\sigma^k$. Compute and output $\mathbf{c} = (\sum_{i=1}^d \alpha_i \mathbf{a}_i + A^* \mathbf{r} + \mathbf{p}\mathbf{e}', \sum_{i=1}^d \alpha_i b_i + \mathbf{r}^T \mathbf{b}^*)$.
- $D_s(\mathbf{c})$: Compute and output $m' = \llbracket \langle \mathbf{c}, \mathbf{s} \rangle \rrbracket_Q$.

The encoding scheme satisfies completeness and IND-CPA security. For clarity, we defer the properties of the encoding scheme in Appendix A.

Zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK)

In this subsection, we present the formal definitions of zk-SNARKs and their properties.

Definition 7 (zk-SNARK) For a relation \mathcal{L} , a zero-knowledge succinct non-interactive argument of knowledge protocol Π comprises three ppt algorithms $(\Pi.\text{Setup}, \Pi.\text{Prove}, \Pi.\text{Verify})$.

1. $(\text{crs}, \text{vrs}, \text{td}) \leftarrow \Pi.\text{Setup}(1^\lambda, u)$: Given the security parameters and a statement u , the setup algorithm generates three components: a common reference string denoted as crs , verification secret information represented by vrs , and the trapdoor denoted as td .
2. $\pi \leftarrow \Pi.\text{Prove}(\text{crs}, u, \omega)$: On receiving u, ω , and crs , the prove algorithm produces a proof π .

3. $0/1 \leftarrow \Pi.\text{Verify}(\text{crs}, \text{vrs}, \pi)$: Taking crs, vrs and π as inputs, the verify algorithm yields a bool symbol 1 or 0 to indicate the acceptance or rejection of the proof.

A zk-SNARK scheme exhibits four fundamental properties, namely completeness, zero-knowledge, argument of knowledge, and succinctness.

Definition 8 (Completeness) For a statement u included in the relation, the setup algorithm outputs $(\text{crs}, \text{vrs}, \text{td}) \leftarrow \Pi.\text{Setup}(1^\lambda, u)$, and the prove algorithm outputs a proof $\pi \leftarrow \Pi.\text{Prove}(\text{crs}, u, \omega)$. If $\Pr[\Pi.\text{Verify}(\text{crs}, \text{vrs}, \pi) = 1] = 1 - \text{negl}(\lambda)$, then Π is complete.

Definition 9 (Zero-knowledge) For any $(u, \omega) \in \mathcal{L}$, a ppt simulator \mathcal{S} exists such that $\{\Pi.\text{Prove}(u, \omega, \text{crs})\} \approx \{\mathcal{S}(u, \text{td})\}$, where $(\text{crs}, \text{vrs}, \text{td}) \leftarrow \Pi.\text{Setup}(1^\lambda, u)$ and \approx can denote perfect, statistically, and computationally indistinguishable. Then this argument system Π is zero-knowledge.

Definition 10 (Argument of Knowledge) For any statement u , and if a ppt adversary can produce a proof π^* passing the verification, then a probabilistic polynomial-time extractor Ext exists and extracts a witness ω satisfying $(u, \omega) \in \mathcal{L}$ with polynomial probability. Equivalently, we have $\Pr[(\pi^*; \omega) \leftarrow (\mathcal{A} \parallel \text{Ext})(\text{crs}, u) \wedge \Pi.\text{Verify}(\text{crs}, \text{vrs}, \pi^*) = 1] = \text{poly}(\lambda)$, where $(\text{crs}, \text{vrs}, \text{td}) \leftarrow \Pi.\text{Setup}(1^\lambda, u)$. Then the non-interactive argument system Π satisfies the argument of knowledge.

Definition 11 (Succinctness) If the argument length of an argument system is sublinear in the security parameter and the circuit size is included in the relation, we say that it is succinct.

Optimization techniques

In this subsection, we present several optimized techniques used in our schemes, including noise smudging, modulus-switching, key-switching, packing, and unpacking.

Noise Smudging. Noise smudging from Gentry (2009) is commonly used to obfuscate additive-homomorphic evaluated ciphertexts or fresh ciphertexts.

Lemma 12 (Noise Smudging, Gentry (2009)) Let B_1, B_2 be positive integers, and k be the statistical security parameter. For an arbitrary integer $m \in [-B_1, B_1]$, we pick n

uniformly at random from the interval $[-B_2, B_2]$. Then if $B_1/B_2 = \text{negl}(k), \{m + n\} \stackrel{s}{\approx} \{n\}$.

Modulus-switching. The modulus-switching technique from Brakerski et al. (2014) can transform a large modulus to a comparatively small modulus without knowing the secret key.

Definition 13 (Modulus-switching) For any integers $k, Q > Q' > p$, and any vector $\mathbf{x} \in R^k, \mathbf{x}' \leftarrow \text{ModSwit}(\mathbf{x}, Q, Q', p)$ is defined as the closest R^k -vector to $\frac{Q'}{Q}\mathbf{x}$ satisfying $\mathbf{x}' = \mathbf{x} \bmod p$.

Lemma 14 (Correctness of modulus switching) Q, Q', p are positive integers satisfying $Q > Q' > p$ and $Q = Q' = 1 \bmod p$. R is a ring with degree n and κ is the statistical parameter. For any $\mathbf{c} \in R^{k+1}$, let $\mathbf{c}' \leftarrow \text{ModSwit}(\mathbf{c}, Q, Q', p)$. Then for any $\mathbf{s} = (-\mathbf{s}', 1)$ with $\mathbf{s}' \leftarrow \Phi_\sigma^k$ satisfying $\|[(\mathbf{c}, \mathbf{s})]_Q\|_\infty < \frac{Q}{2} - \frac{Q'}{2}(n + \sigma\sqrt{nk\kappa})$, the probability of $\|[(\mathbf{c}, \mathbf{s})]_Q\|_p = \|[(\mathbf{c}', \mathbf{s})]_{Q'}\|_p$, and $\|[(\mathbf{c}', \mathbf{s})]_{Q'}\|_\infty < \frac{Q'}{2} \|[(\mathbf{c}, \mathbf{s})]_Q\|_\infty + \frac{p}{2}(n + \sigma\sqrt{nk\kappa})$ is at least $1 - 2n \exp(-\pi\kappa/\sigma^2)$.

Key-switching. The key-switching technique from Brakerski et al. (2014) facilitates the transformation of an encryption under secret key \mathbf{s}_1 to another encryption of the same or related message utilizing a distinct secret key \mathbf{s}_2 with the help of key-switching keys.

Definition 15 (Key-switching) For any vector $\mathbf{x} \in R_Q^k$, we can decompose \mathbf{x} as $\sum_{j=0}^{\log Q-1} \mathbf{y}_j 2^j$, where $\mathbf{y}_j \in R_2^k$ and define $\text{BD}(\mathbf{x}) = (\mathbf{y}_0, \dots, \mathbf{y}_{\log Q-1})$. $\text{PV}(\mathbf{x})$ is defined as $(\mathbf{x}, 2\mathbf{x}, \dots, 2^{\log Q-1}\mathbf{x})$. The key-switching algorithm is presented as follows:

- **SwitKeyGen**($\mathbf{s}_1, \mathbf{s}_2$): Sample $\mathbf{A}' \leftarrow R_Q^{k_1 \log Q \times (k_2-1)}, \mathbf{e}' \leftarrow \Phi_{\sigma'}^{k_1 \log Q}$. Let $\mathbf{s}' \in R^{k_2-1}$ be the residual vector of \mathbf{s}_2 except for the last row. Compute $\mathbf{a}' = -\mathbf{A}'\mathbf{s}' + p\mathbf{e}'$. Output **switkey** = $(\mathbf{A}', \mathbf{a}') + (\mathbf{0}, \text{PV}(\mathbf{s}_1)) \in R_Q^{k_1 \log Q \times k_2}$.
- **KeySwit**(**switkey**, \mathbf{c}): Output **switkey**^T · **BD**(\mathbf{c}).

Lemma 16 (Correctness of key-switching) For any $\mathbf{s}_1 \in R_Q^{k_1}, \mathbf{s}_2 \in R_Q^{k_2}$ with the last coordinate being 1, **switkey** \leftarrow **SwitKeyGen**($\mathbf{s}_1, \mathbf{s}_2$) and $\mathbf{c}_2 \leftarrow$ **KeySwit**(**switkey**, \mathbf{c}_1). Then we have $\langle \mathbf{c}_2, \mathbf{s}_2 \rangle = p \langle \text{BD}(\mathbf{c}_1), \mathbf{e}' \rangle + \langle \mathbf{c}_1, \mathbf{s}_1 \rangle \bmod Q$.

Packing and Unpacking Algorithms. The packing algorithm operates on the message defined over the ring \mathcal{R} by

treating it as several message slots over R . Conversely, the unpacking technique is responsible for successively converting the ciphertext's other slots into the lowest order and extracting the lowest order slot homomorphically. The extraction process is essentially a homomorphic computation of the trace function, which is further addressed by carrying out homomorphic automorphism evaluations. This idea is derived from Halevi and Shoup (2014, 2020).

- **Plaintext encoding:** Given $\mathbf{c}_1, \dots, \mathbf{c}_\xi \in R^{k+1}$, then $\text{Pack}(\mathbf{c}_1, \dots, \mathbf{c}_\xi) = \mathbf{c}_1 + \mathbf{c}_2 x^n + \dots + \mathbf{c}_{\xi-1} x^{n(\xi-1)}$, where n is the dimension of R .
- **Homomorphic plaintext decoding:** Given a key-switching subalgorithm **KeySwit**, the ciphertext $\mathbf{c} \in \mathcal{R}^{k+1}$ and trace homomorphic evaluation keys $\{\mathbf{B}_i\}_{i \in \mathbb{Z}_{2\xi}^*}$, then compute $\mathbf{c}_1 = \sum_{i \in \mathbb{Z}_{2\xi}^*} \text{KeySwit}(\mathbf{B}_i, \tau_i(\mathbf{c}))$, $\mathbf{c}_2 = \sum_{i \in \mathbb{Z}_{2\xi}^*} \text{KeySwit}(\mathbf{B}_i, \tau_i(\mathbf{c} \cdot x^{-n}))$, \dots , $\mathbf{c}_{\xi'} = \sum_{i \in \mathbb{Z}_{2\xi}^*} \text{KeySwit}(\mathbf{B}_i, \tau_i(\mathbf{c} \cdot x^{-(\xi'-1)n}))$ to obtain individual ciphertexts.

At the end of this section, we present a summary of some essential notations in Table 2.

Square span programs over cyclotomic rings

Square span programs (SSPs) were originally introduced by Danezis et al. (2014) as a novel and distinct characterization of the class NP. While all prior works (to our knowledge) considered SSPs over fields, this work generalizes the notion/construction to the setting of rings (particularly the cyclotomic rings). In this way, the underlying mathematical structure of the SSPs can match the one of Ring-LWE (Lyubashevsky et al. 2010),

Table 2 Overview of parameters and notations

Notations	Explanation
$R = \mathbb{Z}[\zeta_{2n}]$	n -th cyclotomic ring
$\mathcal{R} = \mathbb{Z}[\zeta_{16n}]$	$8n$ -th cyclotomic ring
\mathfrak{p}	prime idea of ring R or \mathcal{R}
$n \in \mathbb{Z}$	dimension of ring R
$k \in \mathbb{Z}$	rank of encoding scheme
$k' \in \mathbb{Z}$	rank of the key-switching key
σ	standard deviation
Φ_σ	discrete Gaussian distribution σ over R
Φ'_σ	discrete Gaussian distribution σ over \mathcal{R}
Q	modulus
Q'	switched modulus
p	plaintext modulus
τ	Galois automorphism

yielding much more efficient SNARK constructions (than the plain-LWE-based instantiations).

Definition 17 (Square Span Programs over Rings) A square span program P over the ring R is represented as a polynomial tuple $(l_0(x), \dots, l_m(x), a(x))$ in $R[x]$, where the degree of each $l_i(x)$ is no more than the degree of $a(x)$. The size of P is m , and the degree d equals the degree of $a(x)$. A vector $\mathbf{s} = (s_1, \dots, s_\ell) \in R^\ell$ ($\ell < m$) is accepted by P if and only if there exists another vector $\mathbf{s}' = (s_{\ell+1}, \dots, s_m) \in R^{m-\ell}$ satisfying $a(x)$ divides $(l_0(x) + \sum_{i=1}^m s_i l_i(x))^2 - 1$.

Moreover, if exactly the vectors $\mathbf{s} \in \{0, 1\}^\ell \subset R^\ell$ satisfying $g(\mathbf{s}) = 1$ are accepted, P is said to verify a boolean function g .

The polynomial $((l_0(x) + \sum_{i=1}^m s_i l_i(x))^2 - 1)/a(x)$ is a integer polynomial since $a(x)$ is monic. Below we are going to show that SSPs over rings (some particular cyclotomic rings) can be used to express general NP verifications. We first describe the following corollary about the linearization of logic gates in a boolean circuit in the ring setting, similar to Theorem 2 in Danezis et al. (2014).

Corollary 18 R is a cyclotomic ring. Assume that C is a circuit having m wires and n fan-in 2 gates. For any prime $p \geq 11$, we can compute a matrix–vector pair $(\mathbf{M}, \mathbf{v}) \in \mathbb{Z}_p^{m \times d} \times \mathbb{Z}_p^d$ (with $d = m + n$) from C . Then to show that C is satisfiable over R , equivalently, find a vector $\mathbf{s} \in R_p^m$ such that $\mathbf{sM} + \mathbf{v} \in \{0, 2\}^d$. Moreover, $\mathbf{sM} + \mathbf{v} \in \{0, 2\}^d$, results in $\mathbf{s} \in \{0, 1\}^m$.

Based on this corollary, we can express a boolean circuit C as a ring matrix–vector pair (\mathbf{M}, \mathbf{v}) . Subsequently, we delineate the method for constructing an SSP (over ring R) of C from such a pair.

Construction 19 (Square Span Programs over Ring) R is a cyclotomic ring, and the prime p is larger than 11. Let $R_{[0, \pm 1]}$ denote the subset of R with coefficients within the range of $[0, \pm 1]$. We assume that for every distinct elements x, y from $R_{[0, \pm 1]}$, the difference $x - y$ is invertible modulo pR .

Taking a circuit C with m wires and n fan-in 2 gates as an input, denote $d = m + n$. Subsequently, we can construct a SSP instance as follows:

- Let $(\mathbf{M}, \mathbf{v}) \in \mathbb{Z}_p^{m \times d} \times \mathbb{Z}_p^d$ be the matrix–vector pair as Corollary 18.

- Select distinct r_1, \dots, r_d in $R_{[0, \pm 1]}$ arbitrarily.
- Interpolate polynomials $l_0(x), \dots, l_m(x)$ of degree at most $d - 1$ such that
 - (1) $l_0(r_i) = v_i - 1 \pmod{pR}$ for $i \in [d]$;
 - (2) $l_i(r_j) = \mathbf{M}_{ij} \pmod{pR}$ for $i \in [m], j \in [d]$.
- Set $a(x) = \prod_{i=1}^d (x - r_i)$ and output $(a(x), l_0(x), \dots, l_m(x))$.

We notice that the third step of the above construction is well-defined—any degree $d - 1$ polynomial over $R_p[x]$ (say, $f(x)$) can be uniquely determined given any d values in R_p (say, y_1, \dots, y_d) evaluated at r_1, \dots, r_d . This is because the j -th Lagrange basis polynomial $\ell_j(x) = \prod_{i=1, i \neq j}^d (x - r_i)(r_j - r_i)^{-1}$ is uniquely defined, as every $(r_j - r_i)^{-1}$ (the multiplicative inverse over modulo pR) uniquely exists.

Theorem 20 The prime p satisfies $p \equiv 3 \pmod{8}$, and R is a cyclotomic ring with degree (a power of 2) n . Let $p > 4n$, and $3^n > d$. Then Construction 19 is a square span program over the ring R_p .

Proof Initially, we prove that all the steps involved in Construction 19 are well-defined under the conditions in the theorem statement. Subsequently, we proceed to demonstrate that the output of this construction is an SSP over R_p .

In order to substantiate the well-definedness of the steps, we need to show the following claims: (1) in Step 2, there are indeed d distinct elements in $R_{[0, \pm 1]}$, and (2) in Step 3, the multiplicative inverse (in R_p) of every $(r_i - r_j)$ exists.

Claim (1) is easy to see, as there are 3^d distinct elements in $R_{[0, \pm 1]}$ and $3^n > d$ from the theorem statement. Claim (2) follows from Lemma 21.

Lemma 21 (Katsumata and Yamada 2016) The prime p satisfies $p \equiv 3 \pmod{8}$, and R is a cyclotomic ring with degree (a power of 2) n . Let $p > 4n$. For any distinct element x and y in $R_{[0, \pm 1]}$, the difference $x - y$ is invertible in R_p .

This concludes the first part of our goal. Below we show that the construction outputs an SSP over R_p .

Given the circuit C mentioned above, we can construct a matrix–vector pair $(\mathbf{M}, \mathbf{v}) \in \mathbb{Z}_p^{m \times d} \times \mathbb{Z}_p^d$ as Corollary 18. Proving the circuit C is satisfiable equals that finding a vector $\mathbf{s} \in R_p^m$ such that $\mathbf{sM} + \mathbf{v} \in \{0, 2\}^d$. Moreover, $\mathbf{sM} + \mathbf{v} \in \{0, 2\}^d$ equals $\mathbf{sM} + \mathbf{v} - \mathbf{1} \in \{-1, 1\}^d$, further implying $(\mathbf{sM} + \mathbf{v} - \mathbf{1}) \circ (\mathbf{sM} + \mathbf{v} - \mathbf{1}) = \mathbf{1}$, where \circ denotes entry-wise product and $\mathbf{1}$ is the all-1 vector.

Next, as the construction sets $l_i(r_j) = \mathbf{M}_{ij}$ for $i > 0$ and $l_0(r_j) = \mathbf{v}_j$, the following holds.

$$\begin{aligned} \mathbf{sM} + \mathbf{v} - \mathbf{1} &= (s_1, \dots, s_m) \cdot \begin{pmatrix} l_1(r_1) & \cdots & l_1(r_d) \\ \vdots & \ddots & \vdots \\ l_m(r_1) & \cdots & l_m(r_d) \end{pmatrix} \\ &+ (l_0(r_1), \dots, l_0(r_d)) \\ &= \left(\sum_{i=1}^m s_i l_i(r_1) + l_0(r_1), \dots, \sum_{i=1}^m s_i l_i(r_d) + l_0(r_d) \right). \end{aligned}$$

Thus we obtain the following expression: $(\mathbf{sM} + \mathbf{v} - \mathbf{1}) \circ (\mathbf{sM} + \mathbf{v} - \mathbf{1}) - \mathbf{1} = \left((\sum_{i=1}^m s_i l_i(r_1) + l_0(r_1))^2 - 1, \dots, (\sum_{i=1}^m s_i l_i(r_d) + l_0(r_d))^2 - 1 \right)$.

Given any $\mathbf{s} \in R_p^m$ such that $(\mathbf{sM} + \mathbf{v} - \mathbf{1}) \circ (\mathbf{sM} + \mathbf{v} - \mathbf{1}) - \mathbf{1} = \mathbf{0}$, the equivalent condition is that for every $j \in [d]$, we have $(\sum_{i=1}^m s_i l_i(r_j) + l_0(r_j))^2 - 1 = 0$, meaning that $\{r_j\}_{j \in [d]}$ are the roots of the polynomial $(\sum_{i=1}^m s_i l_i(x) + l_0(x))^2 - 1$. Thus, $a(x) = \prod_{i=1}^d (x - r_i)$ divides $(\sum_{i=1}^m s_i l_i(x) + l_0(x))^2 - 1$.

To conclude, we notice that if C is satisfiable, a vector \mathbf{s} exists such that $(\mathbf{sM} + \mathbf{v} - \mathbf{1}) \circ (\mathbf{sM} + \mathbf{v} - \mathbf{1}) - \mathbf{1} = \mathbf{0}$. The above argument further implies that $a(x)$ divides the polynomial $(\sum_{i=1}^m s_i l_i(x) + l_0(x))^2 - 1$. Conversely, if a vector \mathbf{s} exists to make $a(x)$ divides the polynomial, then $\{r_j\}_{j \in [d]}$ must be the roots of the polynomial, implying $(\mathbf{sM} + \mathbf{v} - \mathbf{1}) \circ (\mathbf{sM} + \mathbf{v} - \mathbf{1}) - \mathbf{1} = \mathbf{0}$. This again proves that C is satisfiable.

Putting things together shows that Construction 19 is a square span program over the ring R_p . \square

Assumptions

The security of previous SNARK schemes relied on two long-standing assumptions: power knowledge of exponent (PKE) assumptions and power Diffie-Hellman (PDH) assumptions.

The PKE assumption, introduced by Gennaro et al. (2013), is a kind of knowledge assumption, which extends the knowledge of exponent assumption (KEA). The original PKE assumption used a discrete logarithm-hard group-based encoding scheme. Later, Gennaro et al. (2018) changed the encoding scheme to LWE-based schemes.

The PDH assumption was proposed by Boneh et al. (2005) and Groth (2010), whose hardness is built on discrete logarithm problems due to the encoding scheme. After altering the encoding scheme directly, Gennaro et al. (2018) obtained new instantiations, whose hardness relies on the LWE problem.

To build our SNARK schemes, it is necessary to broaden the PDH and PKE assumptions in the ring setting. These two assumptions are formally defined in Subsection 4.1. Moreover, we observe a specific scenario in which these assumptions are developed with some useful auxiliary information. The auxiliary information enables us to do ciphertext operations to promote efficiency without harming the hardness of assumptions, which is explained in Subsection 4.2.

Assumptions in the ring setting

The q -PKE assumption and q -PDH assumption in the ring setting follow the nature of those in Gennaro et al. (2013, 2018), except the encoding scheme is instantiated as Module-LWE. The slight modification originates from the structure difference, i.e., group, integer rings, and polynomial rings.

Definition 22 (q -PKE Assumption Over Ring) R is a cyclotomic ring with degree n and prime modulus p . (K, E, D, Eval) is an encoding scheme. The q -PKE assumption over R states that for any ppt adversary \mathcal{A} and some auxiliary information $\text{aux} \in \{0, 1\}^{\text{poly}(\lambda)}$, which is independent of α , there exists a ppt extractor Ext such that

$$\Pr \left[\begin{array}{l} \text{sk} \leftarrow K(1), s, \alpha \leftarrow R_p, \\ \mu = (E_{\text{sk}}(1), E_{\text{sk}}(s), \dots, E_{\text{sk}}(s^q), E_{\text{sk}}(\alpha), E_{\text{sk}}(\alpha s), \dots, E_{\text{sk}}(\alpha s^q)), \\ (c, \hat{c}; a_0, \dots, a_q) \leftarrow (\mathcal{A} \parallel \text{Ext})(pk, \mu, \text{aux}) : \\ D_{\text{sk}}(\hat{c}) = \alpha D_{\text{sk}}(c) \wedge D_{\text{sk}}(c) = \sum_{i=0}^q a_i s^i \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

For the q -PDH assumption in the ring setting, we observe that its form depends on the structure of the ring. Namely, in our choice of ring, R_p is isomorphic to a product of two subfields with norm $p^{n/2}$. A non-zero element $a \in R_p$ means there exists at least one subfield such that a is invertible in the subfield.

Definition 23 (q -PDH Assumption Over Ring) The prime p satisfies $p \equiv 3 \pmod 8$, and R is a cyclotomic ring with degree (a power of 2) n . (K, E, D, Eval) is an encoding scheme. The q -PDH assumption over R is that for any ppt adversary \mathcal{A} ,

$$\Pr \left[\begin{array}{l} \text{sk} \leftarrow K(1), s \leftarrow R_p, \\ \hat{c} \leftarrow \mathcal{A}(E_{\text{sk}}(1), E_{\text{sk}}(s), \dots, E_{\text{sk}}(s^q), E_{\text{sk}}(s^{q+2}), \dots, E_{\text{sk}}(s^{2q})) : \\ D_{\text{sk}}(\hat{c}) \pmod{p_1} \equiv s^{q+1} \text{ or } D_{\text{sk}}(\hat{c}) \pmod{p_2} \equiv s^{q+1} \end{array} \right] \leq \text{negl}(\lambda).$$

that if the adversary wants to utilize the key-switching keys, the remaining part he can do is linear. Then it does not violate the knowledge assumption (PKE assumption).

Next, we give a formal description of the strengthening q -PKE assumption, which embeds proper key-switching keys into the q -PKE assumption:

Definition 24 (The Strengthening q -PKE Assumption) (K, E, D, Eval) is an encoding scheme and $\text{KeySwitch} = (\text{SwitKeyGen}, \text{KeySwit})$ is a key-switching algorithm. The strengthening q -PKE assumption states that for any automorphism or identity mapping f , any ppt adversary \mathcal{A} , any auxiliary information aux and key

switching keys switkey , which are independent of α , there exists a ppt extractor, denoted as Ext , such that

$$\Pr \left[\begin{array}{l} (\text{sk}, F), (\text{sk}', F') \leftarrow K(1), s, \alpha \leftarrow R_p, \text{switkey} \leftarrow \text{SwitKeyGen}(\text{sk}, f(\text{sk}')), \\ \mu = (E_{\text{sk}}(1), E_{\text{sk}}(s), \dots, E_{\text{sk}}(s^q), E_{\text{sk}}(\alpha), E_{\text{sk}}(\alpha s), \dots, E_{\text{sk}}(\alpha s^q)), \\ (c, \hat{c}; a_0, \dots, a_q) \leftarrow (\mathcal{A} \parallel \text{Ext})(\mu, \text{aux}, \text{switkey}, f) : \\ D_{\text{sk}}(\hat{c}) = \alpha D_{\text{sk}}(c) \wedge D_{\text{sk}}(c) = \sum_{i=0}^q a_i s^i \text{ or } D_{\text{sk}'}(\hat{c}) = \alpha D_{\text{sk}'}(c) \wedge D_{\text{sk}'}(c) = \sum_{i=0}^q a_i s^i \end{array} \right] \geq 1 - \text{negl}(\lambda).$$

Assumptions with special auxiliary information

In comparison to the PDH/PKE assumption stated above, we consider a special case where appending some useful auxiliary information. The auxiliary information needs to satisfy the basic principle: admit linear operations only.

Following this idea, we turn a new perspective on the key-switching procedure. As we all know, an integral key-switching algorithm includes two steps: key-switching key generation and the product of bit-decomposed ciphertext and key-switching key. Apparently, the whole key-switching algorithm is non-linear. Nevertheless, with access to the key-switching key, the product can be construed as a linear combination comprising the key-switching key and the decomposition of the ciphertext. Also, no adaptive key-switching keys can be incorporated into the auxiliary information, as the ciphertexts can be evaluated homomorphically by means of modulus-switching and key-switching, as demonstrated in Brakerski et al. (2014).

An important observation is that we can separate the linear and non-linear parts of the key-switching procedure. The separation is putting some predetermined key-switching keys into the auxiliary information. This means

Lemma 25 *If the encoding scheme (K, E, D) satisfies the strengthening q -PKE assumption, then it satisfies the q -PKE assumption over ring.*

Proof The proof is direct. If there is a ppt adversary can break the q -PKE assumption over ring, then it outputs a valid pair (c_1, c_2) such that $D_{\text{sk}}(c_2) = \alpha D_{\text{sk}}(c_1)$ with polynomial probability. This pair is also a valid pair for the strengthening q -PKE assumption. \square

Similarly, we give the formal definition of the strengthening q -PDH assumption.

Definition 26 (The Strengthening q -PDH Assumption) (K, E, D, Eval) is an encoding scheme and $\text{KeySwitch} = (\text{SwitKeyGen}, \text{KeySwit})$ is a key-switching algorithm. The strengthening q -PDH assumption states that for any automorphism or identity mapping f , any ppt adversary \mathcal{A} , any auxiliary information aux and key switching keys switkey ,

$$\Pr \left[\begin{array}{l} (\mathbf{sk}, \mathbf{F}), (\mathbf{sk}', \mathbf{F}') \leftarrow \mathcal{K}(1), s \leftarrow R_p, \text{switkey} \leftarrow \text{SwitKeyGen}(\mathbf{sk}, \mathbf{sk}'), \\ \hat{c} \leftarrow \mathcal{A}(\mathbf{E}_{\mathbf{sk}}(1), \mathbf{E}_{\mathbf{sk}}(s), \dots, \mathbf{E}_{\mathbf{sk}}(s^q), \mathbf{E}_{\mathbf{sk}}(s^{q+2}), \dots, \mathbf{E}_{\mathbf{sk}}(s^{2q}), \text{aux}, \text{switkey}, f) : \\ D_{\mathbf{sk}}(\hat{c}) \bmod p_i \equiv s^{q+1} \text{ for } i = 1 \text{ or } 2 \text{ or } D_{\mathbf{sk}'}(\hat{c}) \bmod p_i \equiv s^{q+1} \text{ for } i = 1 \text{ or } 2 \end{array} \right] \leq \text{negl}(\lambda).$$

Lemma 27 *If the encoding scheme (K, E, D) satisfies the strengthening q -PDH assumption, then it satisfies the q -PDH assumption over ring.*

Proof The proof is similar. If there is a ppt adversary can break the q -PDH assumption over ring, then it outputs an encoding \hat{c} such that $D_{\mathbf{sk}}(\hat{c}) \bmod p_1 \equiv s^{q+1}$ or $D_{\mathbf{sk}}(\hat{c}) \bmod p_2 \equiv s^{q+1}$ with polynomial probability. This encoding is also a valid encoding for the strengthening q -PDH assumption. \square

The Lemmas 25 and 27 show that our new assumptions are stronger than previous ones, which is why it's so named. Next, we give the feasibility of our new assumptions.

Feasibility of New Assumptions. Our modified PKE assumption, which enhances PKE assumption, is rooted in prior knowledge assumptions but refined by the specific ring structure. Furthermore, a set of predetermined key-switching keys is appended to the auxiliary information. The feasibility of this strategy is premised on the key-switching procedure, which can be separated into a non-linear component (Key Generation) and a linear component. Since the key-switching keys are fixed, the adversary is limited to linear evaluations, which does not violate the PKE assumption.

The q -PDH assumption is also amenable to combination with key-switching keys, without compromising the security of the message \mathbf{sk} since the encoding scheme is IND-CPA secure. Consequently, including extra key-switching keys does not impact the difficulty of the q -PDH assumption.

Parameters. The PKE assumption still holds over a small field (or a ring with a small ideal norm). This is due to the sparseness of a valid pair of MLWE encodings, which requires a relation of α between two messages.

Yet, the PDH assumption does not maintain its hardness when considered over a polynomial-sized field \mathbb{F} . The direct consequence is that we can accurately deduce the value of s with a probability of $1/\text{poly}(\lambda)$ and subsequently compute $\mathbf{E}_{\mathbf{sk}}(s^{q+1})$. Moreover, Ishai et al. (2021) proposed a more efficient attack. The adversary can select random and independent $x_1, \dots, x_{2q} \in F$, and compute $f(x) = \prod_{i=1}^{2q} (x - x_i)$, where all x_i are roots of $f(x)$. Then if s collides with any x_i , the adversary can compute $\mathbf{E}_{\mathbf{sk}}(s^{q+1})$ since the coefficient of x^{q+1} in $f(x)$ is not zero with non-negligible probability. Consequently, we require $2q/|\mathbb{F}| < 2^{-\lambda}$ to reach λ -bits security level.

Zero-knowledge succinct non-interactive argument of knowledge schemes

In this section, we present two constructions of zk-SNARKs—one basic construction and then an optimized variant. The basic construction generalizes the framework of SSP-based SNARK (Gennaro et al. 2018) to the ring setting and then applies the technique of modulus switching to reduce the proof length. From the basic scheme, we then design the optimized construction, based on the strengthening assumptions (Definitions 24 and 26) and additional techniques including key-switching and packing, to optimize the parameters.

Below we first present the basic scheme.

The basic scheme

Construction 28 (Basic zk-SANRK) For any NP relation $\mathcal{L} = \{(u, \omega) : C(u, \omega) = 1\}$ related to a boolean circuit C , the protocol Π_1 is composed of three ppt algorithms (Π_1 .Setup, Π_1 .Prove, Π_1 .Verify), and uses an encoding scheme (K, E, D, Eval) (e.g., the Construction 6) and a SSP generation algorithm (e.g., the Construction 19) as building blocks. It works as follows:

- Π_1 .Setup(λ) \rightarrow (crs, vrs, td):
 1. Run $(\mathbf{s}, \mathbf{F}) \xleftarrow{\$} \mathcal{K}(1, k)$ and sample $\beta, r, \alpha \xleftarrow{\$} R_p$. Set $\mathbf{vrs} = \mathbf{td} = (\mathbf{s}, \alpha, \beta, r)$.
 2. Run $\text{ssp} = (a(x), v_0(x), \dots, v_m(x)) \leftarrow \text{SSP}(C)$, and compute $\rho = (\mathbf{E}_s(1), \dots, \mathbf{E}_s(r^d), \mathbf{E}_s(\alpha), \dots, \mathbf{E}_s(\alpha r^d), \mathbf{E}_s(\beta a(r)), \{\mathbf{E}_s(\beta v_i(r))\}_{i=\ell_u+1}^m})$. Set $\mathbf{crs} = (\text{ssp}, \rho, \mathbf{F})$.
 3. Return (crs, vrs, td).
- Π_1 .Prove(crs, u, ω) \rightarrow π :
 1. Parse $u = (u_1, \dots, u_{\ell_u}) \in \{0, 1\}^{\ell_u}$, $\omega = (\omega_{\ell_u+1}, \dots, \omega_m)$, and sample $\gamma \xleftarrow{\$} R_p$. Then compute $v(x) = v_0(x) + \sum_{i=1}^{\ell_u} u_i v_i(x) + \sum_{i=\ell_u+1}^m \omega_i v_i(x) + \gamma a(x)$, $v^*(x) = \sum_{i=\ell_u+1}^m \omega_i v_i(x) + \gamma a(x)$ and $h(x) = (v^2(x) - 1)/a(x)$.
 2. Run Eval to compute
 - $H = \text{Eval}(\{\mathbf{E}_s(r^i), h_i\}_{i=0}^d, \mathbf{F}) = \mathbf{E}_s(h(r))$,

- $\hat{H} = \text{Eval}(\{E_s(\alpha r^i), h_i\}_{i=0}^d, \mathbf{F}) = E_s(\alpha h(r)),$
 - $\hat{V} = \text{Eval}(\{E_s(\alpha r^i), v_i\}_{i=0}^d, \mathbf{F}) = E_s(\alpha v(r)),$
 - $\hat{V}^* = \text{Eval}(\{E_s(\beta v_i(r)), \omega_i\}_{i=\ell_u+1}^m \parallel \{E_s(\beta a(r)), \gamma\}, \mathbf{F}) = E_s(\beta v^*(r)),$
 - $V^* = \text{Eval}(\{E_s(r^i), v_i^*\}_{i=0}^d, \mathbf{F}) = E_s(v^*(r)).$
3. Sample $\{e_{sm,i}\}_{i \in \{1, \dots, 5\}} \xleftarrow{\$} [-B_{sm}, B_{sm}]$, and compute $(H', \hat{H}', \hat{V}', \hat{V}^{*'}, V^{*'}) = (H, \hat{H}, \hat{V}, \hat{V}^*, V^*) + (pe_{sm,1}, pe_{sm,2}, pe_{sm,3}, pe_{sm,4}, pe_{sm,5}).$
 4. Run **ModSwit** to compute
 - $H'' \leftarrow \text{ModSwit}(H', Q, Q', p),$
 - $\hat{H}'' \leftarrow \text{ModSwit}(\hat{H}', Q, Q', p),$
 - $\hat{V}'' \leftarrow \text{ModSwit}(\hat{V}', Q, Q', p),$
 - $\hat{V}^{*''} \leftarrow \text{ModSwit}(\hat{V}^{*'}, Q, Q', p),$
 - $V^{*''} \leftarrow \text{ModSwit}(V^{*'}, Q, Q', p).$
 5. Return $\pi = (H'', \hat{H}'', \hat{V}'', \hat{V}^{*''}, V^{*''}) \in R_Q^{(k+1) \times 5}.$
- $\Pi_1.\text{Verify}(\text{vrs}, u, \tilde{\pi}) \rightarrow 0/1:$
 1. Parse $u = (u_1, \dots, u_{\ell_u}) \in \{0, 1\}^{\ell_u}$, $\tilde{\pi} = (\tilde{H}, \tilde{H}', \tilde{V}, \tilde{V}^*, \tilde{V}^{*'})$ and compute $v_r^* = D_s(\tilde{V}^*), b_r^* = D_s(\tilde{V}^{*'}), h_r = D_s(\tilde{H}), \hat{h}_r = D_s(\tilde{H}'), \hat{v}_r = D_s(\tilde{V}), a_r = a(r)$ and $v_r = v_0(r) + \sum_{i=1}^{\ell_u} u_i v_i(r) + v_r^*.$
 2. Check if the following equations hold:
 - $\alpha h_r = \hat{h}_r,$
 - $\alpha v_r = \hat{v}_r,$
 - $v_r^2 - 1 = h_r \cdot a_r,$
 - $b_r^* = \beta v_r^*.$

If all of the equations are satisfied, then proceed to the subsequent step; otherwise, terminate the process and output “0”.

Theorem 29 *The prime p satisfies $p \equiv 3 \pmod 8$ and the cyclotomic ring R is $\mathbb{Z}[\zeta_{2n}] \cong \mathbb{Z}[X]/(X^n + 1)$ with degree n . Assume the hardness of MLWE assumption, strengthening q -PDH assumption and strengthening q -PKE assumption, as well as IND-CPA security of the encoding scheme. Then for any modulus $Q > 2^{\kappa+4} \sigma n p^2 (d + pn) (p\sqrt{2dn\kappa} + 2\sigma n\kappa)$, $Q = 1 \pmod p$, and the switched modulus $Q' > 4np^2 (\sigma\sqrt{nk\kappa} + n)$, $Q' = 1 \pmod p$, the Construction 28 is a zero-knowledge succinct non-interactive adaptive argument of knowledge (zk-SNARK) for any square span program relation $(u, \omega) \in \mathcal{L}$.*

The proof shares some similarities with the proof of our later optimized proof. For brevity, we defer the proof in Appendix B.

The optimized scheme

The optimized scheme further improves the efficiency of the basic construction using more algebraic techniques— at a high level, we can pack multiple Module-LWE encodings in a lower dimension ring to one Module-LWE encoding in a higher dimension ring, via packing technique. As encodings from a higher dimension ring have a better rate, i.e., output/input length ratio, then the key-switching technique can further compress the length of the proof (by a factor of 8x from our concrete instantiations). However, as the key-switching procedure requires an additional key-switching key, our proof of security would rely on a stronger assumption (Assumptions 26, 24). Below we present the description of the optimized scheme.

Construction 30 (Optimized zk-SNARK) For any NP relation $\mathcal{L} = \{(u, \omega) : C(u, \omega) = 1\}$ related to a boolean circuit C , the optimized protocol Π_2 is composed of three ppt algorithms ($\Pi_2.\text{Setup}, \Pi_2.\text{Prove}, \Pi_2.\text{Verify}$), and uses an encoding scheme (K, E, D, Eval) (e.g., the Construction 6), a SSP generation algorithm (e.g., the Construction 19) and a key switching algorithm (SwitKeyGen, KeySwit) as building blocks. It is defined as follows:

- $\Pi_2.\text{Setup}(\lambda) \rightarrow (\text{crs}, \text{vrs}, \text{td}):$

1. $\text{Run } (\mathbf{s}_1, \mathbf{F}) \leftarrow \text{K}(1, k), (\mathbf{s}_2, \mathbf{F}_2), (\mathbf{s}_3, \mathbf{F}_3) \leftarrow \text{K}(1, k') \text{ independently.}$

Sample $\alpha, \beta, r \xleftarrow{\$} R_p$. Set $\text{vrs} = \text{td} = (\mathbf{s}_1, \mathbf{s}_2, \alpha, \beta, r)$.

2. Run $\text{ssp} = (a(x), v_0(x), \dots, v_m(x)) \leftarrow \text{SSP}(C).$

$\text{Run } \mathbf{B} \leftarrow \text{SwitKeyGen}(\mathbf{s}_1, \mathbf{s}_2),$

$\mathbf{B}' \leftarrow \text{SwitKeyGen}(\mathbf{s}_1, \mathbf{s}_3),$

$\mathbf{B}_i \leftarrow \text{SwitKeyGen}(\tau_i(\mathbf{s}_2), \mathbf{s}_3),$

for $i \in \mathbb{Z}_{16}^*$, where τ_i are pre-determined

automorphisms over R .

Then run E to obtain $\rho = (E_{\mathbf{s}_1}(1), E_{\mathbf{s}_1}(r), \dots, E_{\mathbf{s}_1}(r^d), E_{\mathbf{s}_1}(\alpha), E_{\mathbf{s}_1}(\alpha r), \dots, E_{\mathbf{s}_1}(\alpha r^d),$

$$E_{s_1}(\beta a(r)), \{E_{s_1}(\beta v_i(r))\}_{i=\ell_u+1}^m.$$

$$\text{Set crs} = (ssp, \rho, \mathbf{F}, \mathbf{B}, \mathbf{B}', \{\mathbf{B}_i\}_{i \in \mathbb{Z}_{16}^*}).$$

3. Return (crs, vrs, td).

• $\Pi_2.\text{Prove}(\text{crs}, u, \omega) \rightarrow \pi'$:

1. Parse $u = (u_1, \dots, u_{\ell_u}) \in \{0, 1\}^{\ell_u}$, $\omega = (\omega_{\ell_u+1}, \dots, \omega_m)$, and sample $\gamma \xleftarrow{\$} \mathcal{R}_p$. Then compute $v(x) = v_0(x) + \sum_{i=1}^{\ell_u} u_i v_i(x) + \sum_{i=\ell_u+1}^m \omega_i v_i(x) + \gamma a(x)$, $v^*(x) = \sum_{i=\ell_u+1}^m \omega_i v_i(x) + \gamma a(x)$ and $h(x) = (v^2(x) - 1)/a(x)$.
2. Run Eval to compute

- $H = \text{Eval}(\{E_s(r^i), h_i\}_{i=0}^d, \mathbf{F}) = E_s(h(r))$,
- $\hat{H} = \text{Eval}(\{E_s(\alpha r^i), h_i\}_{i=0}^d, \mathbf{F}) = E_s(\alpha h(r))$,
- $\hat{V} = \text{Eval}(\{E_s(\alpha r^i), v_i\}_{i=0}^d, \mathbf{F}) = E_s(\alpha v(r))$,
- $\hat{V}^* = \text{Eval}(\{E_s(\beta v_i(r)), \omega_i\}_{i=\ell_u+1}^m \| \{E_s(\beta a(r)), \gamma\}, \mathbf{F}) = E_s(\beta v^*(r))$,
- $V^* = \text{Eval}(\{E_s(r^i), v_i^*\}_{i=0}^d, \mathbf{F}) = E_s(v^*(r))$.

3. Sample $\{e_{sm,i}\}_{i \in \{1, \dots, 5\}} \xleftarrow{\$} [-B_{sm}, B_{sm}]$, and compute $(H', \hat{H}', \hat{V}', \hat{V}^*, V^*) = (H, \hat{H}, \hat{V}, \hat{V}^*, V^*) + (pe_{sm,1}, pe_{sm,2}, pe_{sm,3}, pe_{sm,4}, pe_{sm,5})$.
4. Run ModSwit to compute

- $H'' \leftarrow \text{ModSwit}(H', Q, Q', p)$,
- $\hat{H}'' \leftarrow \text{ModSwit}(\hat{H}', Q, Q', p)$,
- $\hat{V}'' \leftarrow \text{ModSwit}(\hat{V}', Q, Q', p)$,
- $\hat{V}^{*''} \leftarrow \text{ModSwit}(\hat{V}^{*'}, Q, Q', p)$,
- $V^{*''} \leftarrow \text{ModSwit}(V^{*'}, Q, Q', p)$.

5.

$$\text{Let } \pi = \text{Pack}(\hat{V}^{*''}, H'', \hat{H}'', V^{*''}, \hat{V}'') \in \mathcal{R}_Q^{k+1}.$$

6.

$$\text{Run KeySwittocomputeandreturn } \pi' = \text{KeySwit}(B, \pi).$$

• $\Pi_2.\text{Vefify}(\text{vrs}, u, \tilde{\pi}) \rightarrow 0/1$:

1. Parse $u = (u_1, \dots, u_{\ell_u}) \in \{0, 1\}^{\ell_u}$ and $\tilde{\pi} = (\tilde{H}, \tilde{\hat{H}}, \tilde{V}, \tilde{V}^*, \tilde{V}^*)$.
2.

$$\text{Compute } m' = D_{s_2}(\tilde{\pi}) \in \mathcal{R}, \text{ and parse } m' \text{ as } (b_r^*, h_r, \hat{h}_r, v_r^*, \hat{v}_r, 0, 0, 0).$$

Then compute $a_r = a(r)$ and $v_r = v_0(r) + \sum_{i=1}^{\ell_u} u_i v_i(r) + v_r^*$.
3. Check if the following equations hold:

- $\alpha h_r = \hat{h}_r$,
- $\alpha v_r = \hat{v}_r$,
- $v_r^2 - 1 = h_r \cdot a_r$,
- $\beta v_r^* = b_r^*$.

If all of the equations are satisfied, then proceed to the subsequent step; otherwise, terminate the process and output “0”.

To show the above Construction 30 is a zk-SNARK, we first prove three separated properties, including completeness, the argument of knowledge, and honest-verifier zero-knowledge respectively, which corresponds to Theorem 31, 32, and 33. Then we put them together and further prove the succinctness property to show the Construction 30 is a zk-SNARK.

Completeness

Theorem 31 *The prime p satisfies $p \equiv 3 \pmod 8$, and R, \mathcal{R} are cyclotomic rings with degree $n, 8n$. For any modulus Q satisfying $Q \equiv 1 \pmod p, Q > 2^{\kappa+3} \cdot 9\sigma np^2(d + pn)$ ($p\sqrt{2dn\kappa} + 2\sigma n\kappa$), and switched modulus Q' satisfying $Q' \equiv 1 \pmod p, Q' > 2np^2 \left[9(\sigma\sqrt{n\kappa} + n) + 18\sigma' \sqrt{(k+1)8n\kappa \log Q'} + 16\sigma'' \sqrt{(k+1)8n\kappa \log Q'} \right]$, the Construction 30 satisfies completeness with probability at least $(1 - 8n \exp(-\pi\kappa/\sigma^2)) \cdot (1 - 16n \exp(-\pi\kappa/\sigma^2))$.*

Proof We demonstrate that the infinite norm of the ultimate noise in π' remains below half of the switched modulus when the prover is in accordance with the protocol. Our analysis will elucidate the evolution of noise throughout each step.

In the setup stage, $\{E_{s_1}(\beta v_i(r))\}_{i=\ell_u+1}^m$ and $E_{s_1}(\beta a(r))$ are computed by additive homomorphic evaluations and we have $B_{\text{crs}} = \sigma p^2 \sqrt{2dn\kappa} + 2p\sigma^2 n\kappa$ with probability $1 - 6n \exp(-\pi\kappa/\sigma^2)$. In the proving stage, we first compute 5 evaluations, and the largest noise growth lies in \hat{V}^* , which is $B_{\hat{V}^*} = B_{\text{crs}}(m - \ell_u + pn) = (\sigma p^2 \sqrt{2dn\kappa} + 2p\sigma^2 n\kappa)(m - \ell_u + pn)$ with probability $1 - 6n \exp(-\pi\kappa/\sigma^2)$. Noise smudging makes the error bound increase to $(2^\kappa + 1)B_{\hat{V}^*}$. Then the infinity norm is less than $(2^\kappa + 1)(\sigma p^2 \sqrt{2dn\kappa} + 2p\sigma^2 n\kappa)(m - \ell_u + pn)$ with probability $1 - 6n \exp(-\pi\kappa/\sigma^2)$. After modulus-switching, the bound $B_{\hat{V}^{*''}}$ is less than $\gamma Q' + \frac{p}{2}(\sigma\sqrt{\kappa n\kappa} + n)$ with probability $1 - 2n \exp(-\pi\kappa/\sigma^2)$ together with

$(2^k + 1)(\sigma p^2 \sqrt{2dn\kappa} + 2p\sigma^2 n\kappa k)(m - \ell_u + pn) + 2dnp^2(m - \ell_u + pn) < \gamma Q$. The packing procedure does not introduce extra noise. Applying key-switching introduces additional noise, $p\langle \text{BD}(\pi), \mathbf{e}' \rangle$, and its infinity norm is no more than $p\sigma' \sqrt{(k+1)8n\kappa \log Q'}$ with probability $1 - 16n \exp(-\pi\kappa/\sigma'^2)$. Since the noise in key-switching key \mathbf{e}' is independent of noise in crs , thus the whole error's infinity norm in the proof is no more than $e_{\pi'} = \gamma Q' + \frac{p}{2}(\sigma \sqrt{\kappa n k} + n) + p\sigma' \sqrt{(k+1)8n\kappa \log Q'}$ with probability $(1 - 8n \exp(-\pi\kappa/\sigma'^2)) \cdot (1 - 16n \exp(-\pi\kappa/\sigma'^2))$.

Therefore, the proof can be decrypted correctly as long as $\gamma Q' + \frac{p}{2}(\sigma \sqrt{\kappa n k} + n) + p\sigma' \sqrt{(k+1)8n\kappa \log Q'} < \frac{Q'}{2}$. \square

Computational Honest-verifier Zero-knowledge

Theorem 32 *Assume the hardness of MLWE assumption, strengthening q-PDH assumption and strengthening q-PKE assumption. Suppose that the encoding scheme is IND-CPA secure. Then for any Q, Q' are defined as Theorem 31, the Construction 30 satisfies computational honest-verifier zero knowledge.*

Proof To establish computational honest-verifier zero-knowledge property, we can construct a ppt simulator $\text{Sim} = (\mathcal{S}_1, \mathcal{S}_2)$ such that the distribution of its output is computationally indistinguishable from the distribution of an honest execution. We divide the whole protocol into three stages. The first stage is the setup phase, the second stage is the first three steps of the prover, and the third stage is the remaining three steps of the prover. The construction of Sim is presented in Fig. 1. From the construction, it differs from the real case in two aspects:

one is that $a(r)$ is always invertible in the simulate case; another is that the simulator encodes messages directly by trapdoor instead of applying additive homomorphic evaluation on crs .

In the first stage, the statistical distance of \mathcal{S}_1 and the real setup algorithm is at most $2/p^{\frac{n}{2}}$ as the probability that a random chosen $a(r) \bmod p_i$ equals 0 is $\frac{2 \cdot p^{n/2} - 1}{p^n} \approx 2/p^{\frac{n}{2}}$. This means that the output distribution of $\mathcal{S}_1(u)$ is statistically close to the output distribution produced by the real setup algorithm.

In the second stage, the simulator and real prover take the output of the first stage as inputs and generate $(H'', \hat{H}'', \hat{V}'', \hat{V}^{*''}, V^{*''})$. In the real protocol, the prover uses re-randomized evaluation (Construction 6) and each encoding consists of two parts e.g., (\mathbf{a}, b) . From the Construction 6, we have \mathbf{a} as a pseudo-random ring vector over $\mathcal{R}^{k'}$, assuming the hardness of the MLWE assumption. After noise smudging, the distribution of b is statistically indistinguishable from the noise distribution by Lemma 12.

In the simulation case, the prover encodes directly using the MLWE encoding scheme. Each encoding consists of two parts e.g., (\mathbf{a}', b') . In the MLWE encoding scheme, \mathbf{a}' is truly random. Thus we have the distribution of \mathbf{a} and the distribution of \mathbf{a}' are computationally indistinguishable. After noise smudging, the distribution of b' is statistically indistinguishable from the noise distribution by Lemma 12. Then the distribution of b and b' are the same.

Up to now, we have proven that two executions are computationally indistinguishable after the first two stages. In the third stage, the simulator and the real prover perform the same modulus switching, pack algorithm, and key-switching, which implies the two distributions are indistinguishable.

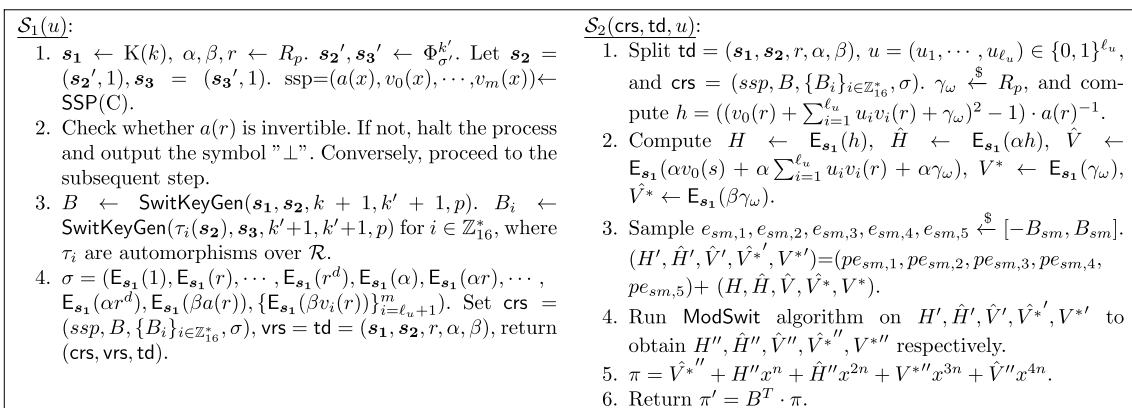


Fig. 1 The construction of simulator $\text{Sim}(u)$

Putting things together, we have that the Construction 30 satisfies computational honest-verifier zero-knowledge. \square

Computational Argument of Knowledge

Theorem 33 *Assume the hardness of MLWE assumption, strengthening q -PDH assumption, and strengthening q -PKE assumption. Suppose that the encoding scheme is IND-CPA secure. Then for any Q, Q' defined as Theorem 31, the Construction 30 satisfies computational argument of knowledge with knowledge error $2(q - m + \ell_u)/p^{\frac{n}{2}}$.*

Proof We show this via a reduction—assuming the existence of a ppt adversary produces a valid proof π' , we can break the hardness of strengthening q -PDH assumption. More concretely, assuming the existence of a ppt adversary, denoted as $\mathcal{A}^{\pi'}$, who can forge a proof for a false statement that passes the verification, it follows that, at least one of the subsequent two events will ensue.

- E_1 : $v^2(r) - 1 = a(r)h(r)$ and $v^2(x) - 1 \neq a(x)h(x)$.
- E_2 : $v^*(x)$ can not be represented as a linear combination of $a(x), v_{\ell_u+1}(x), \dots, v_m(x)$, but the message encoded in the $\hat{V}^{*''}$ equals $\beta v^*(r)$.

We can demonstrate that the occurrence of either event E_1 or E_2 results in breaking the strengthening of q -PDH assumption. The construction of the adversary \mathcal{A}^{PDH} closely resembles that presented in Gennaro et al. (2018). Nevertheless, contrary to the proof presented in Gennaro et al. (2018), our construction is built over the ring. Accordingly, we emphasize the approach to deal with the inverse of a ring element. A valid proof encompasses a single encoding belonging to $\mathcal{R}_{Q'}^{k'+1}$. By executing the unpack algorithm, we obtain 5 encodings. The d -PKE assumption enables the existence of a ppt extractor Ext^{PKE} to extract $h(x)$ from (H'', \hat{H}'') , and $v(x)$ from (V'', \hat{V}'') , where V'' is computed as by homomorphic evaluation and $V^{*''}$. Set $z(x) = v^2(x) - 1 - a(x)h(x)$. The event E_1 implies that $z(x)$ is not zero polynomial and $z(s) = 0$. We assume the highest degree of non-zero coefficient is $k(k \leq 2d)$ and parse $z(x)$ as $\sum_{i=0}^k z_i x^i$. Since $z_k \neq 0$, there exists at least one ideal such that $z_k \bmod p_i \neq 0$ (here z_k is treated as a ring element). We suppose that $z_k \bmod p_1 \neq 0$, and then z_k has its inverse z_k^{-1} in R_p/p_1 without loss of generality.

Next, we show how to compute $E_{s_1}(r^{q+1})$. We have $z(r) \bmod p_1 = 0$ since $z(r) = 0 \bmod p$. Let

$\tilde{z}(x) = ((x^k - z_k^{-1} \cdot z(x)) \bmod p) \bmod p_1$ with degree at most $k - 1$. Clearly, $r^k - \tilde{z}(r)$ equals zero over R/p_1 , so does $r^{q+1} - r^{q+1-k} \tilde{z}(r)$. This means that if we can derive $E_{s_1}(r^{q+1-k} \tilde{z}(r))$, we also obtain $E_{s_1}(r^{q+1})$. As the degree of $x^{q+1-k} \tilde{z}(x)$ is at most q , we compute $E_{s_1}(r^{q+1-k} \tilde{z}(r))$ by homomorphic evaluation $\text{Eval}(\{E_{s_1}(r^{q+1-k+i}), \tilde{z}_i\}_{i=0}^{k-1}, \mathbf{F})$. Furthermore, we require $q \geq 2d - 1$ to make sure $q + 1 - k$ to be positive for k is less than $2d$. This breaks the hardness of strengthening of q -PDH assumption for $q \geq 2d - 1$.

Similarly, if the event E_2 happens, we can also construct an adversary for q -PDH assumption. Specifically, we first generate the crs as the event E_1 happens except the way of computing $\{E_{s_1}(\beta v_i(r))\}_{i=\ell_u+1}^m$ and $E_{s_1}(\beta a(r))$. Similar to the idea of Gennaro et al. (2018), we interpret β as $f(r)$, where $f(x) \in \mathcal{F}$, and \mathcal{F} is defined as the function class: $\{f(x) : \text{the coefficient of } x^{q+1} \text{ in } f(x)v_i(x) \text{ and } f(x)a(x) \text{ are zero, } \forall i \in [\ell_u + 1, m]\}$. In this condition, we generate crs without knowing $E_{s_1}(r^{q+1})$. Meanwhile, the $m - \ell_u + 1$ constraints in \mathcal{F} make the degree freedom of $f(x)$ drop to $q - (m - \ell_u)$. We sample $f(x) \xleftarrow{\$} \mathcal{F}$. Then

$E_{s_1}(\beta v_i(r)) = E_{s_1}(f(r)v_i(r)) = \text{Eval}(\{E_{s_1}(r^j), c_{ij}\}_{j=0, j \neq q+1}^{2q}, \mathbf{F})$
 for $i \in [\ell_u + 1, m]$ and $E_{s_1}(\beta a(r)) = E_{s_1}(f(r)a(r)) = \text{Eval}(\{E_{s_1}(r^j), c'_j\}_{j=0, j \neq q+1}^{2q}, \mathbf{F})$, assuming that $f(x)v_i(x) = \sum_{j=0}^{2q} c_{ij}x^j$ and $f(x)a(x) = \sum_{j=0}^{2q} c'_jx^j$. Similar to the case of event E_1 , we get the proof π' . By running unpacking algorithm on π' , we obtain the separated ciphertexts $(\hat{V}^{*''}, H'', \hat{H}'', V^{*''}, \hat{V}^{*''})$. Next, we prove the coefficient of x^{q+1} in $f(x)v^*(x)$ is invertible (which is treated as a ring element) with overwhelming probability. More specifically, let $f(x) = \sum_{i=0}^q f_i x^i$, $v^*(x) = \sum_{i=0}^d v_i^* x^i$, then $f(x)v^*(x) = \sum_{i=0}^{2d} c_i x^i$ for $q = d$. The coefficient of x^{q+1} in $f(x)v^*(x)$ is $c_{q+1} = \sum_{i=1}^q f_i v_{q+1-i}^*$. We consider the case that c_{q+1} is not invertible, which means that $\sum_{i=1}^q f_i v_{q+1-i}^* = 0 \pmod{p_i}$ for any $i \in \{1, 2\}$. The probability of the case where c_{q+1} is not invertible is at most $2(q - m + \ell_u)/p^{n/2}$ by Schwartz-Zippel lemma. Since the Schwartz-Zippel lemma holds in the field, all elements here are considered as elements in R/p_i . Therefore, the coefficient of x^{q+1} is invertible in R_p with probability $1 - 2(q - m + \ell_u)/p^{n/2}$. Recall that $V^* = E_{s_3}(\beta v^*(r)) = E_{s_3}(f(r)v^*(r)) = E_{s_3}(\sum_{i=0}^{2q} c_i r^i)$. Then we can obtain $E_{s_2}(r^{q+1})$ by $V^{*''}$ subtracts other terms (via homomorphic evaluation and key switching) and multiples c_{q+1}^{-1} . Concretely, we can compute $E_{s_3}(r^{q+1} \bmod p_1) = c_{q+1}^{-1}(V^{*''} - \mathbf{c}')$, where \mathbf{c}' is

$\text{Eval}(\{E_{s_1}(r^i), c_i\}_{i=0, i \neq q+1}^{2q}, \mathbf{F})$ after modulus-switching and key-switching. That breaks the strengthening of q -PDH assumption for $q = d$.

So far, we have established the computational soundness of the proposed Construction 30 with soundness error $2(q - m + \ell_u)/p^{\frac{n}{2}}$. Furthermore, the construction also satisfies the argument of knowledge property, i.e., the existence of a ppt extractor to recover the witness when the adversary outputs convincing proof. As the event E_2 happens with negligible probability, the recovered $v^*(x)$ is a linear combination of $\{a(x), v_{\ell_u+1}(x), \dots, v_m(x)\}$. Then there are $m - \ell_u + 1$ unknowns and $d + 1$ constraints. The witness $\omega = (\omega_{\ell_u+1}, \dots, \omega_m)$ can be recovered easily by Gaussian elimination since $d = m + n > m - \ell_u$. \square

Corollary 34 *Assume the hardness of MLWE assumption, strengthening q -PDH assumption, and strengthening q -PKE assumption. Assume the encoding scheme is IND-CPA secure. Then for any R, \mathcal{R}, p, Q, Q' are defined as Theorem 31, the Construction 30 is a zk-SNARK for any NP relation $(u, \omega) \in \mathcal{L}$.*

Proof To show the Construction 30 is a zk-SNARK, we show four properties, including completeness, the argument of knowledge, honest-verifier zero-knowledge, and succinctness, are satisfied.

Firstly, the succinctness property is evident since the proof consists of a single MLWE encoding, which implies a constant-sized proof and achieves succinctness. From the Theorem 31, we have the Construction 30 satisfies completeness. From the Theorem 32, we have the Construction 30 satisfies computational honest-verifier zero-knowledge. From the Theorem 33, we have the Construction 30 satisfies the computational argument of knowledge.

Put all the pieces together, we prove that the Construction 30 is a zk-SNARK. \square

Concrete parameters

In this section, we exhibit explicit and quantifiable parameters for our basic and optimized schemes.

Parameter selection

Firstly, we summarize the preceding restrictions on parameters and then propose several parameter sets.

- **Message Modulus p :** The choice of p is jointly influenced by the PDH assumption and SSP instance generation. We have opted for a specific scenario where pR is divided into two ideals, and in this case, the prime p satisfies $p \equiv 3 \pmod 8$. To guarantee the robustness of the d -PDH assumption over the subfield R/\mathfrak{p} (where \mathfrak{p} is an ideal of pR) and ensure the accuracy of SSP instance generation over ring R_p , we impose the following requirements: $\log p > 2(\lambda + \log 2d)/n$ and $p > 4n$. After several attempts, we have determined that $n = 64, p = 283$, as well as $n = 32, p = 643$ (for $d = 2^{20}$), or alternatively $n = 32, p = 547$ (for $d = 2^{16}$).
- **Dimension n of R :** The ring dimension n is set to be a power of 2 and it can be small, such as 64, as long as we set a larger rank k to maintain sufficient nk in the MLWE estimation. Analyze with p , and we set $n = 64$ or $n = 32$.
- **Standard deviation σ and σ' :** In this paper, we set all standard deviations $\sigma = \sigma' = 64$ without other annotations.
- **Modulus Q, Q' :** The modulus Q and Q' are positive integers that satisfy completeness of construction as Theorem 31.

Table 3 Parameter setting for $\lambda \approx 128, \kappa = 40$

Scheme	d	p	n	n'	k	k'	$\log Q$	$\log Q'$	Security (Classical)	Security (Quantum)
Basic Scheme	2^{16}	547	32	-	155	-	115	40	128.5	149.3
	2^{16}	283	64	-	77	-	114	40	129.1	149.8
	2^{20}	643	32	-	162	-	120	41	128.2	149.1
	2^{20}	283	64	-	80	-	118	40	129.4	150.1
Optimized Scheme	2^{16}	547	32	256	158	8	117	50	128.8	149.6
	2^{16}	283	64	512	78	4	116	50	128.2	149
	2^{20}	643	32	256	165	8	122	51	128.5	149.4
	2^{20}	283	64	512	81	4	120	50	128.2	149.1

- **Rank k, k' :** The quantities k and k' are measured by the LWE security estimator (Albrecht et al. 2015) for a desired security level given predetermined values $n, \alpha, \sigma, \sigma'$. In terms of classical security, we adopt “ADPS16” (Alkim et al. 2016) method, which yields the least security level relative to other approaches with equivalent parameters. In the case of quantum security, two methodologies, namely “LasMosPol14” (Laarhoven et al. 2015) and “qsieve”, yield identical results.
- **Circuit size d :** We take circuit size ranging from 2^{10} to 2^{20} , which is sufficient in the majority of applications.

Table 4 Proof and CRS lengths of schemes for $\lambda \approx 128, \kappa = 40$

Scheme	d	n	Proof Length	CRS Length (Compressed)
Basic Scheme	2^{16}	32	121.88KB	86.25MB
	2^{16}	64	121.88KB	171MB
	2^{20}	32	130.53KB	1.41GB
	2^{20}	64	126.56KB	2.77GB
Optimized Scheme	2^{16}	32	14.06KB	133.99MB
	2^{16}	64	15.63KB	246.94MB
	2^{20}	32	14.34KB	1.48GB
	2^{20}	64	15.63KB	2.88GB

* For CRS length, we merely count encodings, and other parts, including the seed for PRF are neglected

Following the aforementioned parameter suggestions, we present detailed parameters for partial circuits ($d = 2^{16}$ and $d = 2^{20}$ as before) in Table 3.

Proof and CRS length

The proof of the basic scheme consists of 5 encodings in \mathcal{R}_Q and that in the optimized scheme is 1 encoding in \mathcal{R}_Q . Then the proof size of the basic scheme and optimized scheme are $5n(k + 1) \log Q'$ bits, and $n'(k' + 1) \log Q'$ bits respectively. For the basic scheme, CRS consists of $2(d + 1) + m - \ell_u + 3$ encodings in \mathcal{R}_Q^{k+1} , which are less than $3(d + 1)(k + 1)n \log Q$ bits. Furthermore, we can utilize a seed and a pseudorandom generator to substitute true randomness in the encodings, then the length of CRS shrinks to $3(d + 1)n \log Q$ bits. Since the optimized scheme utilizes the key-switching technique, the CRS length in the optimized scheme increases by key-switching keys. To be specific, the optimized scheme employs 2 key-switchings from \mathcal{R}_Q^{k+1} to $\mathcal{R}_Q^{k'+1}$ and 8 key-switchings from $\mathcal{R}_Q^{k'+1}$ to \mathcal{R}_Q^{k+1} , which are $8n(k' + 1)(2(k + 1) + 8(k' + 1)) \log^2 Q'$ bits.

Plug the estimated values into the formulae, we obtain the concrete proof and CRS lengths in Table 4 and depict the tendency for circuit size ranging from 2^{10} to 2^{20} in Figs. 2 and 3.

Comparison Between the Basic and the Optimized Schemes. As shown in Figs. 2 and 3, our results indicate a slight increase in the proof length alongside a nearly linear increase in the CRS length. (It is important to

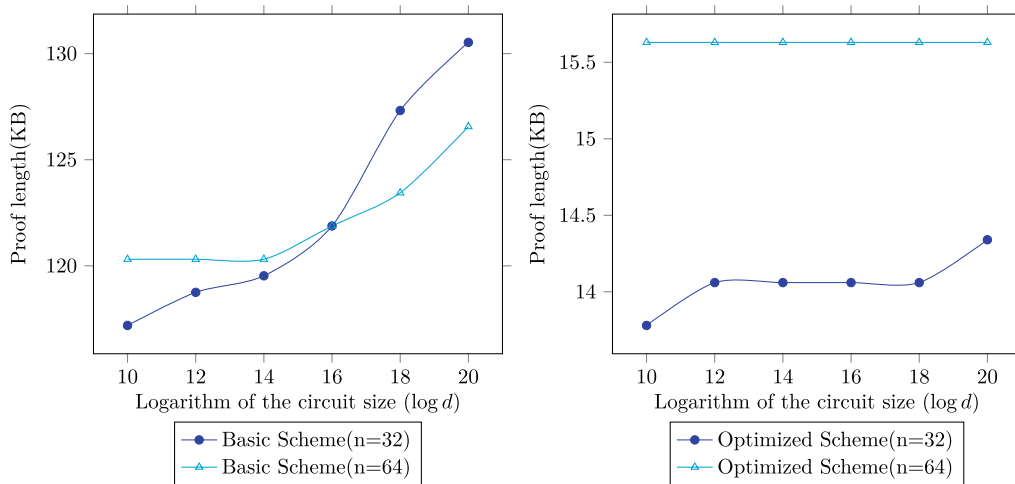


Fig. 2 Proof length varying from circuit size

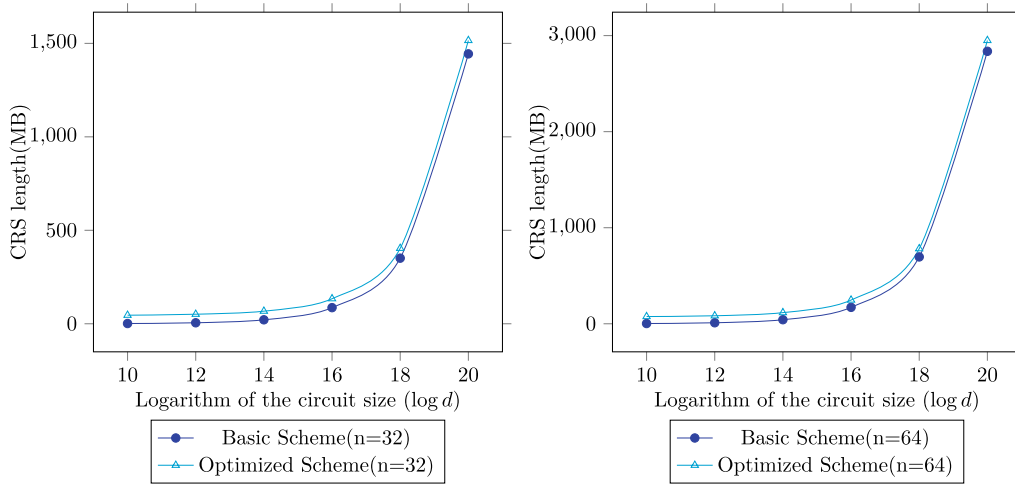


Fig. 3 CRS length varying from circuit size

note that our horizontal axis is logarithmic in scale with respect to circuit size, which is why the growth follows an exponential pattern.) This is due to the slight effect of circuit size on switched modulus, which translates to a small impact on proof length. Conversely, the increase in circuit size has a significant impact on the CRS length, which displays an almost linear correlation.

Our optimized scheme offers a marked improvement over the basic scheme, with the proof length being roughly 5x shorter. This attributes to its single encoding, as opposed to the basic scheme’s five encodings. As for the CRS length, the difference between the two schemes is minimal, primarily arising from the size of key-switching keys, which constitutes only 1% of the total CRS size at $d = 2^{20}$.

Conclusion

In this paper, we develop the framework of square span program-based SNARKs and design new zk-SNARKs over cyclotomic rings. To fit in the ring setting, we first extend square span programs over rings and then propose two new assumptions. Based on these fundamental components, we construct SANRKs by applying module-switching and key-switching procedures in a novel way.

Our scheme avoids parallel repetition leveraging the ring structure. Thus, we obtain concretely small constructions for SNARKs with the designated verifier in the preprocessing model, which has a proof of length 14.06KB and a CRS of length 133.99MB for the circuit of size 2^{16} . For larger circuits, i.e., the size of 2^{20} , the proof length and CRS length of our scheme are 14.34KB and 1.48GB respectively. These are 23.3% smaller and the CRS length is 3.6x smaller compared to those in Ishai et al. (2021).

Appendix A: Properties of encoding scheme

Lemma A.1 (Correctness) *The encoding scheme $(K, E, D, Eval)$ is defined in the Construction 6. Then for any d independent encodings $E_{sk}(m_1), \dots, E_{sk}(m_d)$ and any $\alpha_i \in R_p$, the infinity norm of error in $Eval(E_{sk}(m_i), \alpha_i, F)$ is no more than $\sigma p^2 \sqrt{dn\kappa} + 2p\sigma^2 n\kappa k$ with probability $1 - 6n \exp(-\pi\kappa/\sigma^2)$.*

Proof As the decoding algorithm depicts, we have $\sum_{i=1}^d \alpha_i b_i + \mathbf{r}^T \mathbf{b}^* - \langle \sum_{i=1}^d \alpha_i \mathbf{a}_i + A^* \mathbf{r} + p\mathbf{e}', \mathbf{s}' \rangle = \sum_{i=1}^d \alpha_i \cdot p\mathbf{e}_i + \mathbf{r}^T \cdot p\mathbf{e}^* - p\langle \mathbf{e}', \mathbf{s}' \rangle$. Assume that e_i is the error in $E_{sk}(m_i)$ and e_{ik} is the k -th bit representation of e_i for $i \in [d], k \in [n]$. a_{ik} is defined in similar manner. Since the e_{ik} are independent, every entry of noise in $\sum_{i=1}^d \alpha_i e_i$ is a linear combination of e_{ik} . Take the constant term as an example, it equals $\sum_{i=1}^d \sum_{k=0}^{n-1} \alpha_{ik} p e_{i,n-k}$, which is bounded by $p\sigma \sqrt{\kappa} \|\alpha\|_2 \leq p\sigma \sqrt{\kappa} \sqrt{dn\kappa} = \sigma p^2 \sqrt{dn\kappa}$ with probability at least $1 - 2 \exp(-\pi\kappa/\sigma^2)$ by Lemma 1. An element sampled from Φ_σ is bounded by $\sigma \sqrt{\kappa}$ with probability at least $1 - 2 \exp(-\pi\kappa/\sigma^2)$, and two independent elements multiplied is no more than $\sigma^2 \kappa$ with probability $1 - 4 \exp(-\pi\kappa/\sigma^2)$, thus the infinity norm of $\mathbf{r}^T \mathbf{e}^*$ is no more than $\sigma^2 \kappa k$ with probability $1 - 4n \exp(-\pi\kappa/\sigma^2)$. The bound of $\langle \mathbf{e}', \mathbf{s}' \rangle$ is estimated as well. According to the union bound, the infinity norm of $Eval(E_{sk}(m_i), \alpha_i)$ is no more than $\sigma p(p\sqrt{dn\kappa} + 2\sigma n\kappa k)$ with probability at least $1 - 6n \exp(-\pi\kappa/\sigma^2)$. \square

Lemma A.2 (Security) *Let n, k, Q, σ be as defined in Construction 6. Then the Construction 6 is CPA-security under the hardness of MLWE assumption.*

Appendix B: Proofs of the basic scheme

Proof To prove the Construction 28 is a zk-SNARK, we need to prove its four properties: completeness, computational soundness, argument of knowledge, and succinctness. Firstly, the succinctness property is satisfied as the proof is constant, i.e., 5 encodings. Next, we show the remaining three properties.

Completeness. If all infinite norms of the accumulated noise in the encodings contained in the proof π are smaller than half of the switched modulus, the descriptions can be performed by the verifier correctly. Then the completeness property is satisfied. Next, we analyze the noise generated in each step.

In the setup stage, $\{E_s(\beta v_i(r))\}_{i=\ell_u+1}^m$ and $E_s(\beta a(r))$ are computed by additive homomorphic evaluations and we have $B_{\text{crs}} = \sigma p^2 \sqrt{2dnk} + 2p\sigma^2 nkk$ with probability $1 - 6n \exp(-\pi k/\sigma^2)$. In the proving stage, we first compute 5 evaluations, and the largest noise growth lies in \hat{V}^* , which is $B_{\hat{V}^*} = B_{\text{crs}}(m - \ell_u + pn) = (\sigma p^2 \sqrt{2dnk} + 2p\sigma^2 nkk)(m - \ell_u + pn)$ with probability $1 - 6n \exp(-\pi k/\sigma^2)$. Noise smudging makes the error bound increase to $(2^\kappa + 1)B_{\hat{V}^*}$. Then the infinity norm is less than $(2^\kappa + 1)(\sigma p^2 \sqrt{2dnk} + 2p\sigma^2 nkk)(m - \ell_u + pn)$ with probability $1 - 6n \exp(-\pi k/\sigma^2)$. After modulus-switching, the bound $B_{\hat{V}^*}$ is less than $\gamma Q' + \frac{p}{2}(\sigma \sqrt{knk} + n)$ with probability $1 - 2n \exp(-\pi k/\sigma^2)$ together with $(2^\kappa + 1)(\sigma p^2 \sqrt{2dnk} + 2p\sigma^2 nkk)(m - \ell_u + pn) + 2dnp^2(m - \ell_u + pn) < \gamma Q$.

Let $\sigma = \alpha Q$, and the parameter α represents the error rate. In addition, we take $\gamma = 1/8np$. By approximate scaling, we have $Q > 2^{\kappa+4} \sigma np^2 (d + pn) (p\sqrt{2dnk} + 2\sigma nkk)$, and $Q' > 4np^2 (\sigma \sqrt{knk} + n)$.

Computational Honest-Verifier Zero-knowledge. The analysis can be regarded as a simplified version of the proof of Theorem 32. To avoid repetitions, we stress the difference instead of repeating the whole process.

In the first stage, the setup algorithm just consists of encodings of $1, r, \dots, r^d, \alpha, \dots, \alpha r^d, \beta a(r), \beta v_m(r), \dots, \beta v_{\ell_u+1}(r)$. The simulator for this stage removes other keys as well and checks $a(r)$ whether is invertible, implying statistically indistinguishability with statistical difference $2/p^{n/2}$. In the second stage, the proof is exactly the same, and two distributions are computationally

indistinguishable. In the third stage, the prover only considers the modulus-switching process. Then two distributions are computationally indistinguishable.

Computational Argument of Knowledge The proof is also included in the proof for Theorem 33. The key difference is without the unpacking algorithm and all secret keys are s . The details are omitted. \square

Acknowledgements

Not applicable.

Author contributions

All the authors have equal contributions to this paper.

Funding

This work is supported by the National Key R&D Program of China under Grant 2020YFA0712303. Zhedong Wang is supported by National Natural Science Foundation of China (Grant No.62202305) and Shanghai Pujiang Program under Grant 22PJ1407700.

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 30 July 2023 Accepted: 29 January 2024

Published online: 19 March 2024

References

- Albrecht MR, Player R, Scott S (2015) On the concrete hardness of learning with errors. *J Math Cryptol* 9(3):169–203
- Albrecht MR, Cini V, Lai RW, Malavolta G, Thyagarajan SA (2022) Lattice-based snarks: publicly verifiable, preprocessing, and recursively composable. In: Annual international cryptology conference. Springer, pp 102–132
- Alkim E, Ducas L, Pöppelmann T, Schwabe P (2016) Post-quantum key exchange: a new hope. In: 25th USENIX security symposium (USENIX Security 16), pp 327–343
- Banaszczyk W (1995) Inequalities for convex bodies and polar reciprocal lattices in \mathbb{R}^n . *Discrete Comput Geom* 13:217–231
- Ben-Sasson E, Chiesa A, Genkin D, Tromer E, Virza M (2013) Snarks for c: verifying program executions succinctly and in zero knowledge. In: Advances in cryptology—CRYPTO 2013: 33rd annual cryptology conference, Santa Barbara, CA, USA, August 18–22 2013. Proceedings, Part II. Springer, pp 90–108
- Ben-Sasson E, Chiesa A, Tromer E, Virza M (2014) Succinct {Non-Interactive} zero knowledge for a von Neumann architecture. In: 23rd USENIX security symposium (USENIX Security 14), pp 781–796
- Bitansky N, Canetti R, Chiesa A, Tromer E (2011) From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. *Cryptology ePrint Archive*
- Bitansky N, Canetti R, Chiesa A, Tromer E (2012) From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Proceedings of the 3rd innovations in theoretical computer science conference, pp.326–349
- Bitansky N, Chiesa A, Ishai Y, Paneth O, Ostrovsky R (2013) Succinct non-interactive arguments via linear interactive proofs. In: Theory of cryptography: 10th theory of cryptography conference, TCC 2013, Tokyo, Japan, March 3–6 2013. Proceedings. Springer, pp 315–333

- Bitansky N, Canetti R, Chiesa A, Goldwasser S, Lin H, Rubinfeld A, Tromer E (2017) The hunting of the snark. *J Cryptol* 30(4):989–1066
- Boneh D, Boyen X, Goh EJ (2005) Hierarchical identity based encryption with constant size ciphertext. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, pp 440–456
- Boneh D, Ishai Y, Sahai A, Wu DJ (2017) Lattice-based snarks and their application to more efficient obfuscation. In: annual international conference on the theory and applications of cryptographic techniques. Springer, pp 247–277
- Bonneau J, Meckler I, Rao V, Shapiro E (2020) Coda: decentralized cryptocurrency at scale. *Cryptology ePrint Archive*
- Brakerski Z, Gentry C, Vaikuntanathan V (2014) (Leveled) fully homomorphic encryption without bootstrapping. *ACM Trans Comput Theory (TOCT)* 6(3):1–36
- Chiesa A, Yaguev E (2020) Barriers for succinct arguments in the random oracle model. In: Theory of cryptography: 18th international conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part II 18. Springer, pp 47–76
- Chung H, Kim D, Kim JH, Kim J (2023) Amortized efficient zk-snark from linear-only rlwe encodings. *J Commun Netw*
- Cini V, Lai RW, Malavolta G (2023) Lattice-based succinct arguments from vanishing polynomials. In: Annual international cryptology conference. Springer, pp 72–105
- Danezis G, Fournet C, Groth J, Kohlweiss M (2014) Square span programs with applications to succinct nizk arguments. In: International conference on the theory and application of cryptology and information security. Springer, pp 532–550
- Fisch B, Liu Z, Vesely P (2023) Orbweaver: succinct linear functional commitments from lattices. In: Annual international cryptology conference. Springer, pp 106–131
- Gennaro R, Gentry C, Parno B, Raykova M (2013) Quadratic span programs and succinct nizks without pcps. In: Advances in Cryptology—EUROCRYPT 2013: 32nd annual international conference on the theory and applications of cryptographic techniques, Athens, Greece, May 26–30 2013. Proceedings 32. Springer, pp 626–645
- Gennaro R, Minelli M, Nitulescu A, Orrù M (2018) Lattice-based zk-snarks from square span programs. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 556–573
- Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: Proceedings of the forty-first annual ACM symposium on theory of computing, pp 169–178
- Gentry C, Wichs D (2011) Separating succinct non-interactive arguments from all falsifiable assumptions. In: Proceedings of the forty-third annual ACM symposium on theory of computing, pp 99–108
- Goldwasser S, Micali S, Rackoff C (1989) The knowledge complexity of interactive proof systems. *SIAM J Comput* 18(1):186–208
- Goldwasser S, Lin H, Rubinfeld A (2011) Delegation of computation without rejection problem from designated verifier cs-proofs. *Cryptology ePrint Archive*
- Groth J (2010) Short pairing-based non-interactive zero-knowledge arguments. In: Advances in cryptology-ASIACRYPT 2010: 16th international conference on the theory and application of cryptology and information security, Singapore, December 5–9 2010. Proceedings 16. Springer, pp 321–340
- Groth J (2016) On the size of pairing-based non-interactive arguments. In: *Advances in Cryptology—EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Vienna, Austria, May 8–12, 2016, Proceedings, Part II 35, pp. 305–326. Springer
- Halevi S, Shoup V (2014) Algorithms in Helib. In: Advances in cryptology—CRYPTO 2014: 34th annual cryptology conference, Santa Barbara, CA, USA, August 17–21, 2014, Proceedings, Part I 34. Springer, pp 554–571
- Halevi S, Shoup V (2020) Design and implementation of helib: a homomorphic encryption library. *Cryptology ePrint Archive*
- Ishai Y, Su H, Wu DJ (2021) Shorter and faster post-quantum designated-verifier zk-snarks from lattices. In: Proceedings of the 2021 ACM SIGSAC conference on computer and communications security, pp 212–234
- Katsumata S, Yamada S (2016) Partitioning via non-linear polynomial functions: more compact ibes from ideal lattices and bilinear maps. In: Advances in cryptology—ASIACRYPT 2016: 22nd international conference on the theory and application of cryptology and information security, Hanoi, Vietnam, December 4–8 2016, Proceedings, Part II 22. Springer, pp 682–712
- Laarhoven T, Mosca M, Van De Pol J (2015) Finding shortest lattice vectors faster using quantum search. *Des Codes Crypt* 77:375–400
- Labs P (2018) Filecoin. <https://filecoin.io/filecoin.pdf>
- Langlois A, Stehlé D (2015) Worst-case to average-case reductions for module lattices. *Des Codes Crypt* 75(3):565–599
- Lyubashevsky V, Peikert C, Regev O (2010) On ideal lattices and learning with errors over rings. In: Advances in cryptology—EUROCRYPT 2010: 29th annual international conference on the theory and applications of cryptographic techniques, French Riviera, May 30–June 3, 2010. Proceedings 29. Springer, pp 1–23
- Naganuma K, Yoshino M, Inoue A, Matsuoka Y, Okazaki M, Kunihiro N (2020) Post-quantum zk-snark for arithmetic circuits using gaps. In: 2020 15th Asia joint conference on information security (AsiaJICIS). IEEE, pp 32–39
- Naor M (2003) On cryptographic assumptions and challenges. In: Annual international cryptology conference. Springer, pp 96–109
- Nitulescu A (2019) Lattice-based zero-knowledge snarks for arithmetic circuits. In: Progress in cryptology—LATINCRYPT 2019: 6th international conference on cryptology and information security in Latin America, Santiago de Chile, Chile, October 2–4, 2019, Proceedings 6. Springer, pp 217–236
- Parno B, Howell J, Gentry C, Raykova M (2016) Pinocchio: nearly practical verifiable computation. *Commun ACM* 59(2):103–112
- Peikert C, Vaikuntanathan V, Waters B (2008) A framework for efficient and composable oblivious transfer. In: Annual international cryptology conference. Springer, pp 554–571
- Peikert C, Pepin Z, Sharp C (2021) Vector and functional commitments from lattices. In: Theory of cryptography: 19th international conference, TCC 2021, Raleigh, NC, USA, November 8–11 2021, Proceedings, Part III 19. Springer, pp 480–511
- Sasson EB, Chiesa A, Garman C, Green M, Miers I, Tromer E, Virza M (2014) Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE symposium on security and privacy. IEEE, pp 459–474
- Wee H, Wu DJ (2023) Succinct vector, polynomial, and functional commitments from lattices. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, pp 385–416

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.